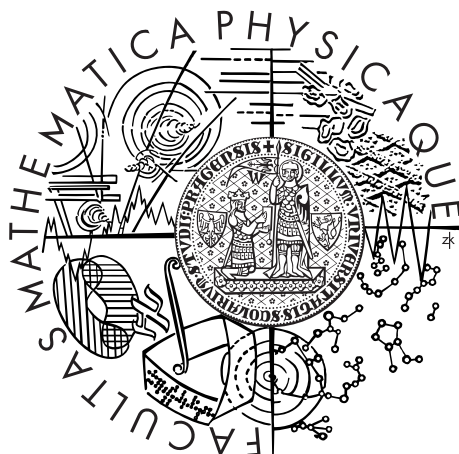


Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Peter Hrinčár

AudioCam

Kabinet software a výuky informatiky

Vedoucí bakalářské práce: RNDr. Jan Horáček

Studijní program: Informatika

Studijní obor: obecná informatika

Praha 2012

Touhle cestou bych rád poděkoval všem, kteří mě podporovali při psaní této bakalářské práci. Obzvlášť bych rád poděkoval svému vedoucímu práce RNDr. Janovi Horáčkovi, za odborné vedení a cenné rady, které mi poskytl při její tvorbě.

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V dne

Podpis autora

Název práce: AudioCam

Autor: Peter Hrinčár

Katedra: Kabinet software a výuky informatiky

Vedoucí bakalářské práce: RNDr. Jan Horáček

Email vedoucího: jan.horacek@gmail.com

Email autora: peter.hrincar@gmail.com

Abstrakt: Práca sa zaoberá problémom vytvárania hudby pohybom. Naším cieľom bolo vyvinúť software, ktorý umožní užívateľovi ovládať virtuálny hudobný nástroj pomocou webovej kamery. Zameriavame sa hlavne na univerzálne rozhranie ovládané pohybom. S využitím rozdielovej metódy analýzy pohybu a optického toku získavame informácie o pohybe. Virtuálny nástroj na základe týchto informácií vytvára tóny s použitím technológie MIDI. Konkrétne simulujeme virtuálne klávesy, na ktorých užívateľ hrá. Náš návrh časti virtuálneho nástroja bol stavaný tak, aby sa v budúcnosti dal ľahko rozšíriť o ďalšie prístupy pri tvorbe hudby.

Klíčová slova: zpracování pohybu, hudba, webcam

Title: AudioCam

Author: Peter Hrinčár

Department: Department of Software and Computer Science Education

Supervisor: RNDr. Jan Horáček

Email of supervisor: jan.horacek@gmail.com

Email of author: peter.hrincar@gmail.com

Abstract: The thesis deals with the problem of creating music through movement. Our aim is to implement software that allows the user to control a virtual musical instrument using a webcam. We focus primarily on universal interface controlled by movement. Using the differential motion analysis method and optical flow we obtain the information about a movement. Based on this information the virtual instrument creates tones using MIDI technology. Specifically, we simulate a virtual keyboard on which the user can play. We designed the virtual instrument in such a way that it could be easily extended in the future with other approaches to the process of creating music.

Keywords: motion processing, music, webcam

Obsah

Úvod	3
1 Súčasný stav poznania	5
1.1 Vývoj nástrojov ovládaných pohybom	5
1.2 Práce podobného zamerania	6
1.3 Umelé produkovanie hudby	7
1.4 Ovládanie počítača webovou kamerou	8
2 Základné poznatky	9
2.1 Webová kamera	9
2.2 Spracovanie digitálneho obrazu	9
2.2.1 Digitalizácia obrazu	10
2.2.2 Predspracovanie obrazu	10
2.3 Úvod do počítačového videnia	12
2.3.1 Segmentácia obrazu	12
2.4 Analýza pohybu	13
2.4.1 Rozdielové metódy detekcie pohybu	13
2.4.2 Pozadie scény	14
2.4.3 Optický tok	14
2.4.4 Detekcia významných bodov	15
2.5 Technológia MIDI	16
3 Analýza problému a návrh riešenia	18
3.1 Podrobný popis problému	18
3.2 Dekompozícia problému	19
3.3 Ovládanie pomocou pohybu	20
3.3.1 Aplikácia rozdielovej metódy	20
3.3.2 Optický tok	22
3.4 Virtuálny nástroj	23
4 Implementácia	24
4.1 Použité technológie	24
4.1.1 Programové prostredie	24
4.1.2 Podporné knižnice	24
4.1.3 Zostavenie projektu	24
4.2 Štruktúra projektu	24
4.3 Dekompozícia úlohy	25
4.4 Popis tried a objektov programu	25

5	Užívateľská dokumentácia	29
5.1	Inštalácia a spustenie programu	29
5.2	Ovládanie programu a popis užívateľského rozhrania	29
5.3	Nastavenia, kalibrácia programu	32
5.4	Riešenie problémov	35
	Záver	36
	Zoznam použitej literatúry	37
A	Obsah priloženého CD	39
B	MIDI	40
B.1	Nástroje General MIDI 1	40
B.2	Mená a čísla MIDI nôt	42

Úvod

Hudobným nástrojom môže byť čokoľvek, čo dokáže vyprodukovať zvuk a zároveň je nejakým spôsobom kontrolovaný hudobníkom. V dnešnej dobe elektronickej hudby sa mení pohľad na hudobné nástroje. Nové technológie ponúkajú nové možnosti aj v tejto oblasti. Hranie na klasické akustické nástroje si vyžaduje určitú zručnosť. Človek musí obetovať hodiny tréovania, aby sa na hudobný nástroj dokázal naučiť hrať. Komponovanie vlastnej hudby si vyžaduje navyše aj stupeň kreativity.

S hudbou sa často spája tanec. Tanec môžeme chápať ako formu vyjadrenia pocitu. Pomocou rytmických pohybov tela dokáže prirodzenou formou reagovať na hudbu. Predstavme si ale, že by sa hudba vytvárala iba na základe pohybu, bez nutnosti akéhokoľvek dotyku. Rozhraním hudobného nástroja by sa stalo celé ľudské telo. Človek by dokázal prirodzene skrz pohyb vyjadriť svoju náladu a pocity hudbou bez toho, aby musel na tento nástroj hodiny trénovať.

Pri takejto odvážnej predstave vzniká niekoľko otázok. Ako a aká hudba by sa vytvárala? Ako by sa hudba kontrolovala, aký vplyv by mal pohyb na tento proces? Akým spôsobom by bolo ľudské telo snímané a aké parametre pohybu by sa získavali?

Prácou sa budeme snažiť zodpovedať na tieto otázky a priblížiť sa o krok k takémuto nástroju. Naším vstupom bude obyčajná webová kamera, ktorá je dnes bežnou dostupnou súčasťou osobných počítačov. Budeme sa z nej snažiť v reálnom čase získať a spracovať maximum možných informácií s použitím poznatkov najmä z oboru počítačového videnia. Jednou z priorít bude, aby sa užívateľ nemusel nijako prispôbovať, teda mohol program zapnúť kedykoľvek a kdekoľvek. Zameriame sa hlavne na rozhranie virtuálneho nástroja ovládaného pohybom a možnosti vytvárania hudby budeme riešiť iba okrajovo. Budeme však skúmať, ako je možné aplikovať získané parametre pohybu pri tvorbe hudby. Pri návrhu budeme hľadiť aj na budúce možné rozšírenie, napríklad v spolupráci s umelcami a hudobnými skladateľmi.

Cieľom práce teda bude vytvoriť program, AudioCam, interaktívny nástroj na produkovanie hudby pomocou webovej kamery. V tejto aplikácii by si mal užívateľ hudbu skladať pohybom rúk bez toho, aby sa musel niečoho dotýkať. Zloženú hudbu si bude môcť nahráť a uložiť. V práci si nekladieme za cieľ vytvoriť program pre profesionálnych hudobníkov. Budeme sa hlavne zaoberať možnosťami spracovania obrazu, detekcie a analýzy pohybu a jeho aplikácii pri tvorbe hudby. Zameriame sa na vytvorenie rozhrania ovládaného pohybom, kde využijeme rozdielovú metódu detekcie pohybu a analýzu optického toku. Práca bude demonštrovať možnosti takéhoto ovládania virtuálneho hudobného nástroja určeného hlavne pre zábavu. K produkovaniu hudby využijeme technológiu MIDI.

Štruktúra práce

Práca je rozdelená do nasledujúcich kapitol:

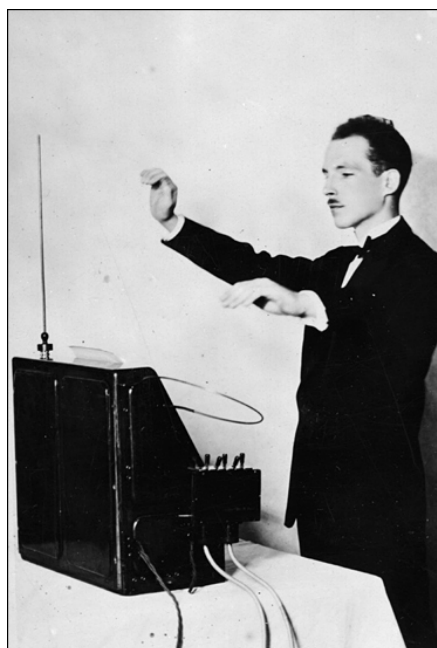
- **Súčasný stav poznania**, v ktorej sa budeme snažiť popísať prehľad problematiky hudobných nástrojov založených na pohybe. Začne s krátkym priezrom históriou a vývojom v tejto oblasti. Ďalej budú nasledovať príklady prác podobného zamerania. Stručne spomenieme aj problém umelého vytvárania hudby a následne sa budeme zaoberať časťou problému spojeného s ovládaním počítača webovou kamerou.
- **Základné poznatky**, kde sa pokúsime uviesť čitateľa do kontextu práce cez základné poznatky o webovej kamere, teoretické podklady z oblasti spracovania obrazu, počítačového videnia a technológie MIDI. Niektoré poznatky z tejto kapitoly budú priamo aplikované pri riešení problému a budeme sa na ne odkazovať v ďalších kapitolách.
- **Analýza problému a návrh riešenia** sa snaží analyzovať problém vytvorenia virtuálneho nástroja ovládaného prostredníctvom webovej kamery. Najskôr začneme podrobným popisom problému. Následne si problém rozdelíme na časti a potom navrhujeme celkové riešenie. Popíšeme riešenie ovládacieho rozhrania virtuálneho hudobného nástroja. Budeme sa snažiť vytvoriť univerzálne rozhranie použiteľné pre viac rôznych typov nástrojov, kde použijeme rozdielovú metódu a analýzu optického toku. Ďalej navrhujeme virtuálny hudobný nástroj, ktorý bude simulovať hranie na klávesy a budeme sa zaoberať budúcimi rozšíreniami programu o ďalšie prístupy pri tvorbe hudby.
- **Implementácia** popisuje použité technológie, štruktúru a zostavenie projektu. Obsahuje prehľad dôležitých algoritmov, tried a objektov programu.
- **Užívateľská dokumentácia**, kde sa nachádza postup pri inštalácii programu, popis užívateľského rozhrania a ovládania programu. Ďalej spôsoby kalibrácie a riešenia možných problémov.

Kapitola 1

Súčasný stav poznania

1.1 Vývoj nástrojov ovládaných pohybom

Myšlienka vytvoriť hudobný nástroj ovládaný pohybom nie je nijako nová. Už v roku 1919 sa sovietskemu vedcovi Levu Sergejevičovi Tereminovi podarilo vytvoriť Theremin [16]. Je to svetovo prvý elektronický hudobný nástroj, na ktorý sa hrá bez toho, aby sa ho človek dotýkal. Tento nástroj si našiel po celom svete veľké množstvo svojich fanúšikov a to i napriek tomu, že hranie na neho je nesmierne náročné.



Obr. 1.1: Profesor Lev Teremin demonštruje jeho Theremin.
Zdroj : www.freeinfosociety.com

Ako môžeme vidieť na obrázku 1.1, Theremin je vlastne oscilátor, drevená skrinka s dvoma anténami. Človek pomocou pravej ruky kontroluje frekvenciu, výšku tónu. Pomocou ľavej amplitúdu, hlasitosť. Na prvých Thereminoch sa dali ovplyvňovať iba tieto základné vlastnosti tónu. Moderné modely dnes umožňujú meniť tiež farbu zvuku. Za spomenutie určite stojí aj Thereminov ďalší objav, Rhytmicon, prvý elektronický rytmický generátor. AudioCam by mohol v istej

miere Theremin simulovať, avšak budeme sa snažiť o flexibilnejšie využitie.

Po úspechu Thereminu nasledovali ďalšie pokusy spojiť hudbu a pohyb, ktoré nadviazali na podobné princípy. Príkladom je umelec John Cage a jeho piata časť zo svojej série prác z polovice šesťdesiatych rokov, označenej tiež Variations V [17]. Podarilo sa mu premeniť tanečný parket na alternatívny hudobný nástroj. Použil senzory umiestnené na javisku, antény reagujúce na pohyb, pomocou ktorých sa produkoval zvuk. Tanečníci tak vytvárali svojim tancom hudbu, priamo závislú na ich polohe a pohyboch.

Vývoj digitálnej techniky a nových technológií zohral zásadnú úlohu aj v oblasti rozvoja počítačovej hudby a virtuálnych hudobných nástrojov založených na pohybe. S príchodom senzorov určených na detekciu a spracovanie pohybu a výkonných počítačov tak priniesol ďalšie možnosti. Otázkou však ostalo, ako údaje o pohybe interpretovať do sveta hudby, pretože existujúce modely akustických nástrojov na ne zjavne nebolo možné jednoducho aplikovať. Umelci v spolupráci s programátormi sa spolu začali venovať tomuto problému.

1.2 Práce podobného zamerania

Zaujímavá z nášho hľadiska je medzinárodná konferencia Nime¹, ktorá je venovaná vedeckému výskumu vývoja nových technológií zameraných na hudobný prejav a umelecké stvárnenie [18]. Od roku 2001 sa každoročne zídu umelci z celého sveta a podelia sa o svoje nápady a riešenia nových hudobných rozhraní. Častými témami sú napríklad v reálnom čase gestami ovládaná hudba, návrh, či dizajn nástrojov. Nachádza sa tu veľká rôznorodosť v prezentovaných prácach, ktorá je zapríčinená neobmedzenými možnosťami, používanými vstupmi, metódami a technikami mapovania vstupov do výstupov a podobne. Z nášho pohľadu nás zaujímajú práce v skupine kamerou ovládaných hudobných nástrojov. Spomeňme tiež medzinárodnú konferenciu počítačovej hudby ICMC² [19], ktorá taktiež prináša inovatívne práce zaoberajúce sa aj touto témou.

Príkladom práce prezentovanej aj na konferencii Nime je Iamascope [5]. Ide o prácu zaradenú do skupiny kamerou ovládaných hudobných nástrojov. Autom sa podarilo vytvoriť interaktívny kaleidoskop, ktorý prináša unikátny audiovizuálny zážitok. Skladá sa z dvoch subsystémov, grafického a hudobného. Užívateľ v reálnom čase kontroluje hudbu pomocou pohybov jeho tela, pričom pred sebou vidí kaleidoskopický obraz. Z kamerového snímku sa použije koláčový výsek, segment na vytvorenie kaleidoskopického obrazu. Napríklad, ak sa použije 20 stupňový výsek, potom na vytvorenie kaleidoskopu treba 18 týchto segmentov. Výsek sa ďalej rozdelí na niekoľko častí. Pre každú časť sa určí priemerná hodnota rozdielu intenzít z dvoch po sebe idúcich snímok. Ak táto hodnota prekročí určitý prah, užívateľ vykonal pohyb v danom mieste, zahrá sa určitý tón. Hodnota tónu závisí od danej časti, v ktorej pohyb nastal. Každá časť má určený relatívny posun od aktuálneho tónu, ktorý určuje systém. Aktuálne tóny sa periodicky vyberajú zo sekvencie a menia sa za určitý časový interval. Pri vhodnom výbere tejto sekvencie je výsledkom melodická hudba. AudioCam bude vychádzať z podobného princípu zachytenia pohybu. V riešení sa však zameriavame na

¹New Interfaces for Musical Expression

²International Computer Music Conference

rozhranie, ktoré sa neskôr bude dať rozšíriť aj o nové prístupy pri tvorbe hudby.

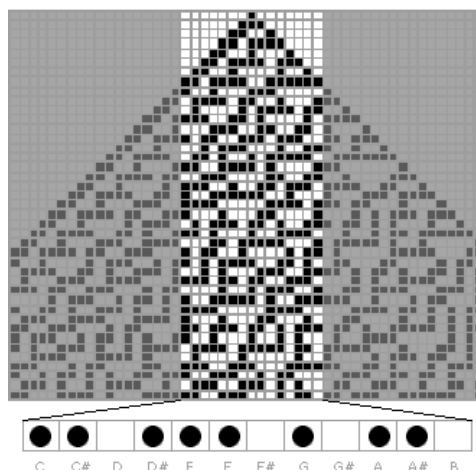
V ďalšej zaujímavej práci [6] taktiež experimentujú s generovaním hudby, ktorá je iniciovaná ľudským pohybom. Výsledkom je systém, ktorý vytvára tóny na základe zmeny rýchlosti, zrýchlenia a polohy používateľových končatín a kĺbov. V reálnom čase sleduje celé telo pomocou použitia senzoru Microsoft KinectTM pre Xbox 360. Práca sa skladá z aplikácie na zachytávanie pohybu a aplikácie na generovanie hudby. Aplikácia na zachytávanie pohybu zabezpečuje komunikáciu medzi senzorom KinectTM a počítačom. Získava informácie o polohe časti tela v trojrozmernom priestore. Pomocou Open Sound Control (OSC) knižnice komunikuje s aplikáciou na generovanie hudby. Pri generovaní hudby využívajú Max/MSP, grafický objektovo orientovaný jazyk, obecné určený pre interaktívne vytváranie multimédií. Celkovo sa snažili prispôsobiť audio výstup tak, aby umelec ľahko priradil zvuky k pohybu jeho tela. V našej práci, ktorá okrem iného získava iba dvojrozmerné dáta, sa viac ako na polohu zameriavame na pohyb samotný. Ich práca a mnohé ďalšie, ktoré využívajú senzor Kinect, sú inšpiráciou pre budúce pokračovanie našej práce.

1.3 Umelé produkovanie hudby

Predchádzajúce príklady prác jednoduchým spôsobom mapovali pohyb do tónov. Samozrejme existujú aj komplikovanejšie mapovania, ktoré využívajú sofistikovanejšie metódy. Pomocou pohybu kontrolujú určité parametre, vlastnosti, ktoré sú aplikované na produkovanie hudby. Môže to byť napríklad tempo, rytmus alebo melódia. Aby ale bolo možné aplikovať tieto vlastnosti, je nutné mať predstavu, ako hudbu na základe týchto vlastností vytvoriť. Môžu sa použiť preddefinované skladby, alebo sa môže hudba generovať umelo. Automatické generovanie hudby je netriviálny problém, ktorým sa naša práca zaoberá iba okrajovo.

Jeden z mnohých prístupov ako tento problém riešiť môžeme vidieť v projekte WolframTones [20], dostupného online. Vychádza zo základného objavu Stephena Wolframa [7], pomocou jednoduchých pravidiel alebo programov je možné dosiahnuť veľkej komplexnosti. Tento fakt objavil na pokusoch s jednorozmernými celulárnymi automatmi. Toto zistenie sa aplikuje aj pri tvorbe hudby. Základný princíp jednorozmerného celulárneho automatu je veľmi jednoduchý. Pracuje sa s riadkom buniek s hodnotami 0 alebo 1. Ďalej sú dané pravidlá, ktoré určujú hodnotu bunky v nasledujúcom kroku výpočtu, v závislosti od jej aktuálnej hodnoty a hodnoty jej okolitých susedov. Niektoré pravidlá vytvoria jednoduché štruktúry, v iných je možné nájsť pozoruhodné zložité a náhodné vzory. Tie sa tak môžu použiť aj pri vytvorení jedinečnej skladby. Pretože tento systém sleduje určité pravidlá, vytvorí sa skladba s vnútornou konzistenciou, čo je asi to, čo z toho robí hudbu. WolframTones používa rôzne algoritmy na vytvorenie hudby zo vzorov získaných výpočtom celulárneho automatu, príklad je vidieť na obrázku 1.2.

Tento spôsob je ale príklad extrémneho prístupu, pretože generovaná hudba je vopred určená jasnými pravidlami. Zmenou týchto pravidiel, alebo prístupu pri vytváraní hudby na základe parametrov by sme však mohli dosiahnuť dynamicky sa meniacu skladbu, reagujúcu na pohyb. Síce sa v práci budeme zaoberať hlavne problémom rozhrania virtuálneho nástroja ovládaného pohybom, jeho neskoršie rozšírenie môže fungovať aj na podobnom princípe.



Obr. 1.2: Obrázok znázorňuje zjednodušený princíp fungovania WolframTones. História behu výpočtu celulárneho automatu, z ktorého výrez sa použije pri tvorbe hudby. Bunky tohoto výrezu sú napríklad mapované do tónov hudobnej stupnice. Zdroj: tones.wolfram.com/

1.4 Ovládanie počítača webovou kamerou

Časť problému, ktorým sa práca zaoberá, môžeme zovšeobecniť na ovládanie počítača pomocou webovej kamery. Táto téma je v súčasnej dobe stále viac a viac populárna. Mnohí autori sa ňou zaoberajú a prinášajú svoje pohľady do riešenia tohoto problému. Ako príklad uvedieme čerstvú prácu Richarda Jedličku [10]. V nej je vidieť posun vpred oproti starším akademickým prácam [11] [12], ktoré používali špeciálnu rukavicu s LED diódami a mali pevne definované užívateľské gestá. Jeho práca sa zaoberá ovládaním počítača kamerou a snaží sa nahradiť všetky ostatné vstupy gestami vykonávanými za pomoci dobre rozoznateľných jedno-farebných objektov. Tieto gestá si užívateľ definuje podľa svojej potreby. Práca sa blíži ku komerčným softvérom, príkladom ktorého je Camspace [21]. V platenej verzii programu umožňuje plne nahradiť myš, joystick a zároveň podporuje ovládanie pre viacerých užívateľov.

My sa na rozdiel od ovládania počítača snažíme priblížiť k ovládaniu hudby pohybom a preto sa nemusíme sústreďovať na presné nahradzovanie existujúcich vstupov, pri ktorých sa vyžaduje maximálna presnosť. Oproti sledovaniu konkrétnych objektov, alebo pevne určitých zvýraznených bodov, sa zaujímame primárne o pohyb užívateľa, s ktorým ďalej pracujeme pri produkovani hudby.

Kapitola 2

Základné poznatky

Predtým, ako sa pustíme do analýzy problému, prejdeme si základné poznatky týkajúce sa danej problematiky.

2.1 Webová kamera

Od čias, kedy začala v roku 1991 prvá webová kamera snímať kávovú kanvicu [22], sa toho dosť zmenilo a našla si široké všestranné využitie v rôznych oblastiach. Spočiatku slúžila najmä na prenos snímok z verejných miest prostredníctvom internetu. Neskôr sa s čoraz väčšou dostupnosťou začala využívať hlavne na komunikáciu medzi ľuďmi. Videohovory a konferencie si niektorí bez nej už ani nevedia predstaviť. Webová kamera sa postupne stala bežným doplnkom osobných počítačov. Začala sa integrovať do notebookov a moderných obrazoviek. Dnes sa pomocou vhodného softvéru používa napríklad aj ako vstupné ovládacie zariadenie počítača, alebo ako alternatívny spôsob ovládania hier. Najmä v oblasti herných konzol sa vyvinuli špecializované kamery ponúkajúce napríklad vysoké rozlíšenie alebo 3D snímky. My sa však zameriame na bežne dostupné externé webové kamery, či tie integrované v notebookoch.

Technológia

Štandardná webová kamera sa skladá z objektívu, snímacieho prvku a podpornej elektroniky. Objektív môže byť plastový s možnosťou ručného nastavenia, alebo bez nej s pevným zaostrením. Používa sa taktiež objektív s automatickým zaostrovaním. Snímače môžu byť CMOS alebo CCD. Väčšina webových kamier poskytuje VGA (640×480) rozlíšenie pri frekvencii pohybujúcej sa okolo 30 snímok za sekundu. Táto frekvencia väčšinou ale závisí od svetelných podmienok, preto aj pri používaní AudioCamu musíme zaistiť dobre osvetlené a stabilné prostredie.

2.2 Spracovanie digitálneho obrazu

Nasledujú základné poznatky z oblasti spracovania digitálneho obrazu, čerpané hlavne z odborných literatúr [4][1], ktoré zaistia konzistentnosť pojmov. Vybranú teóriu prejdeme od digitalizácie obrazu až po jeho predspracovanie.

2.2.1 Digitalizácia obrazu

Najskôr si stručne zhrnieme, ako vznikne po snímaní digitálny obraz. Digitalizácia obrazu je proces, pri ktorom dochádza k prechodu od spojitkej funkcie k diskretnej. Môžeme si to predstaviť ako prechod analógovej informácie do digitálneho tvaru. Tento proces sa rozdeľuje na vzorkovanie a kvantovanie. Kvantovanie je diskreditácia oboru hodnôt obrazovej funkcie a proces vzorkovania určuje jej definičný obor. Webová kamera nám teda sprostredkuje digitálny obraz, s ktorým ďalej pracujeme.

Obrazová funkcia

Šedo-tónový obraz, obraz v odtieňoch šedej, budeme vyjadrovať ako dvojrozmernú jasovú funkciu

$$f(x, y).$$

Definičným oborom bude

$$R = \{(x, y), x \geq 0, x \leq mx, y \geq 0, y \leq my\},$$

kde x, y sú celé čísla, mx a my sú maximálne súradnice. Obor hodnôt je množina jasových hodnôt.

Digitálny obraz sa bežne reprezentuje ako matica, ktorej prvky sú obrazové elementy, **pixely**. Najčastejšie 8-bitový (256 grey-scale image), každý pixel v obrázku je uložený ako číslo medzi 0 až 255, kde 0 reprezentuje čierny pixel, 255 biely, hodnoty medzi predstavujú odtiene sivej. Čiernobiely, **binárny obraz**, má iba dve možné hodnoty pre každý obrazový bod, čiernu a bielu. **Farebný obraz** môžeme definovať ako súbor niekoľkých obrazových funkcií, ktoré reprezentujú farebné zložky v závislosti od zvoleného farebného modelu.

Základné farebné modely

RGB¹ je aditívny farebný model, farby vznikajú kombináciou červenej, zelenej a modrej. Farebný model **HSV**² lepšie odpovedá ľudskému vnímaniu farieb. Skladá sa z troch zložiek, ktoré sú farebný tón, sýtosť farby a hodnota jasu. **Farebná hĺbka** popisuje počet bitov k popisu farby bodu.

2.2.2 Predspracovanie obrazu

Po snímaní a digitalizácii nasleduje predspracovanie obrazu. Vstupom a výstupom je obraz. Ide o základný krok, ktorý pomáha zlepšiť obraz z hľadiska ďalšieho spracovania. Napríklad potlačí deformácie a skreslenia, či zvýrazní určité dôležité črty. Zoskupuje operácie s obrazom na najnižšej úrovni abstrakcie, pracuje s maticami, ktoré obsahujú informácie o veľkosti jasu v každom obrazovom bode.

Základné rozdelenie podľa veľkosti okolia spracovaného pixlu:

- bodové jasové transformácie

¹RGB = Red, Green, Blue

²HSV = Hue, Saturation, Value

- geometrické transformácie (teoreticky bod, prakticky jeho malé okolie)
- lokálne predspracovanie
- globálne

Bodové jasové transformácie

Ide o štandardné operácie aplikované na obraz. Výstupná hodnota je vždy vypočítaná na základe bodu vstupného obrazu. Vstupný a výstupný obraz má preto rovnakú veľkosť. Môžeme ich rozdeliť do dvoch skupín:

- modifikácia jasovej stupnice
- jasové korekcie

V modifikácii jasovej funkcie sa jasová hodnota vo vstupnom obraze transformuje bez ohľadu na pozíciu v obraze. Táto metóda sa používa napríklad na zvýšenie kontrastu pomocou ekvalizácie histogramu.

Pri druhom type závisí hodnota jasu v bode výstupného obrazu od jasu na rovnakej pozícii bodu vstupného obrazu. **Operácie** $+$ a $-$ sú najzákladnejšie operácie spracovania obrazu, pomocou nich sčítame alebo odčítame dve hodnoty jasu obrazových bodov vstupných obrazov, alebo jasovú hodnotu zmeníme o konštantu.

$$\begin{aligned}h(x, y) &= f(x, y) \pm g(x, y) \\h(x, y) &= f(x, y) \pm C\end{aligned}$$

f, g sú vstupné obrazové funkcie, h je výstupná, C je konštanta. Podobne môžeme definovať aj ostatné **aritmetické operácie** ako \times a \div .

Geometrické transformácie

Geometrické transformácie sa používajú pre škálovanie (zmenšenie alebo zväčšenie obrazu), posúvanie, otáčanie, skosenie obrazov.

Lokálne predspracovanie

Lokálne operácie predspracovania používajú malé okolie vstupného bodu obrázku na výpočet jeho novej hodnoty. Rozdeľujeme ich z hľadiska použitia:

- vyhladzovanie, snaha o potlačenie šumu
- ostrenie, detekcia hrán

Podľa matematických vlastností ich môžeme rozdeliť na:

- **lineárne metódy**, výsledná jasová hodnota $g(i, j)$ je lineárnou kombináciou jasových hodnôt okolitých bodov okolia \mathcal{O} bodu vstupného obrázku $f(i, j)$.

$$g(i, j) = \sum_{m, n \in \mathcal{O}} h(i - m, j - n) f(m, n),$$

čo je vlastne diskrétna konvolúcia, kde h je konvolučná maska (jadro). Často sa používajú obdĺžnikové masky s nepárnym počtom riadkov a stĺpcov, reprezentatívny bod sa tak môže nachádzať v strede konvolučnej masky. Konvolučné lineárne operácie sa používajú na vyhladzovanie, ostrenie alebo detekciu hrán.

- **nelineárne metódy** na výpočet používajú iba okolité body s určitými vlastnosťami.

Priemerovacie filtre patria medzi vyhladzovacie filtre, ktoré sa snažia potlačiť šum. Pri jednoduchom priemerovaní získame hodnotu jasú ako aritmetický priemer jasových hodnôt okolitých bodov. Prípadne môžeme zväčšiť váhu stredného bodu masky alebo jeho susedov, aby lepšie aproximovali vlastnosti šumu s Gaussovským rozdelením. Koefficienty v konvolučnej maske budú odpovedať 2D Gaussovskému rozdeleniu

$$g(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{\sigma^2}},$$

kde je σ stredná smerodajná odchýlka definujúca veľkosť okolia bodu. Príklady masiek:

$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, h = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}, h = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Mediánový filter je nelineárna metóda, jasová hodnota bodu je určená ako medián z hodnôt jasú bodov v lokálnom okolí bodu vstupného obrazu, napríklad s rozmermi 3×3 . Táto metóda je vhodná na potlačanie impulzného šumu.

2.3 Úvod do počítačového videnia

Počítačové videnie je odbor, ktorého hlavným cieľom je vytvorenie modelu reálneho sveta a získavania použiteľných informácií pomocou zachyteného obrazu [3]. Takouto informáciou je aj pohyb, preto sa na neho bude v tejto práci pozeráť z pohľadu odboru počítačového videnia.

2.3.1 Segmentácia obrazu

Segmentácia obrazu je jedným z najdôležitejších krokov vedúcej k analýze spracovaných obrazových dát. Jej hlavným cieľom je rozdeliť obraz na časti, ktoré majú silnú koreláciu s predmetmi alebo oblasťami reálneho sveta obsiahnuté v obrázku [1].

Existujú rôzne metódy segmentácie, ktoré môžeme v zásade rozdeliť na:

- metódy, ktoré sú založené na obrazových bodoch
- metódy, ktoré sú založené na oblastiach

Popíšeme prahovanie, metódu založenú na hodnotení podľa jednotlivých bodov.

Prahovanie

Prahovanie, thresholding, je jedna z najstarších metód segmentácie obrazu, ktorá je založená na hodnotení jasú každého bodu. Väčšina prahovacích metód predpokladá, že objekty v obraze sa od pozadia dajú rozlíšiť na základe jasovej hodnoty.

Základné prahovanie nahor, pri ktorom sa najskôr zvolí prah T , podľa ktorého sa segmentuje obraz.

$$f(x, y) = \begin{cases} 0 & \text{pre } f(x, y) < T \\ 1 & \text{pre } f(x, y) \geq T \end{cases}$$

Na vstupe máme obraz v odtieňoch šedej a výstupom je (segmentovaný) binárny obraz. Taktiež môžeme uvažovať prahovanie nadol, kde hodnota pixlov segmentovaných prvkov obrazu je menšia ako prah T .

Pri poloprahovaní sa menia len jasové hodnoty väčšie alebo menšie ako prah T . Pri spektrálnom alebo multispektrálnom prahovaní môžeme miesto konkrétnej hodnoty prahu T použiť interval alebo množinu intervalov. Poznáme taktiež adaptívne prahovanie, pri ktorom sa hodnota vypočítava zvlášť pre každý pixel.

2.4 Analýza pohybu

Analýza pohybu je zložitá úloha, preto musíme uvažovať o určitých predpokladoch, ktoré pomôžu znížiť komplexnosť problému. Neexistuje žiaden univerzálny algoritmus, ktorý by fungoval vo všetkých podmienkach.

Detekcia pohybu je tu základný problém, jej časté použitie je napríklad k bezpečnostným účelom. Ďalšia množina problémov je detekcia a rozpoznanie pohybujúcich sa objektov, určenie ich polohy a rýchlosti. Najzložitejšie je analyzovať pohyb, keď sa pohybuje kamera. Našou veľkou výhodou je tak použitie stacionárnej kamery.

2.4.1 Rozdielové metódy detekcie pohybu

Jednoduchú detekciu pohybu môžeme zistiť využitím rozdielu medzi obrazmi snímanými v rôznych časových okamihoch, pričom predpokladáme stacionárnu kameru pri konštantnom osvetlení. Výsledok však vysoko závisí od kontrastu objektu a pozadia.

Nech f_1 a f_2 sú dva po sebe snímané obrazy, potom nazveme ich rozdielový obraz d , pričom hodnota 0 v tomto obraze bude znamenať, že nedošlo k výraznej zmene jasovej úrovne medzi zodpovedajúcimi miestami týchto po sebe idúcich obrazov.

$$d(i, j) = \begin{cases} 0 & \text{pre } |f_1(i, j) - f_2(i, j)| \leq \epsilon \\ 1 & \text{inak,} \end{cases},$$

kde ϵ je malé kladné číslo.

Hodnota 1 v rozdielovom obraze mohla vzniknúť buď

1. $f_1(i, j)$ je bod pohybujúceho sa objektu,
 $f_2(i, j)$ je bod pozadia,
alebo opačne

2. $f_1(i, j)$ je bod pohybujúceho sa objektu,
 $f_2(i, j)$ je bod iného pohybujúceho sa objektu
3. $f_1(i, j)$ je bod pohybujúceho sa objektu,
 $f_2(i, j)$ je bod inej časti rovnakého objektu
4. vplyvom šumu, rôznymi nepresnosťami pri snímaní kamery

Použitím rozdielovej metódy ale nezískame informáciu o smere, miere a trajektórii pohybu. Taktiež sa stráca pohyb pomalých pohybov a malých objektov. Tieto nevýhody však môžeme čiastočne odstrániť použitím metódy **kumulovaného rozdielového obrazu**. Prvý snímok je označený za referenčný a na všetkých nasledujúcich obrazoch sa prevedie operácia rozdielu. Výsledný kumulovaný obraz je súčet všetkých rozdielových snímok, ktoré však môžu mať rôzne váhy.

$$d_{kum}(i, j) = \sum_{k=1}^n a_k |f_0(i, j) - f_k(i, j)|,$$

kde d_{kum} je výsledný kumulovaný obraz, f_0 referenčný obraz a f_k nasledujúce obrazy a ich váhy a_k .

2.4.2 Pozadie scény

V určitých prípadoch je dobré odlišiť objekty od statického pozadia. Definujeme binárny obraz $d(i, j)$, masku rozdeľujúcu obraz na pozadie a popredie.

$$d(i, j) = \begin{cases} 0 & \text{pre } |f(i, j) - b(i, j)| \leq \epsilon \\ 1 & \text{inak} \end{cases},$$

kde $f(i, j)$ je aktuálny snímok a $b(x, y)$ je obrazová funkcia modelu pozadia, ϵ určuje prah. Pretože sa nachádza v snímku určitá miera šumu, na získanie pozadia je lepšie namiesto jedného snímku čistého pozadia použiť sériu snímok a následne vypočítať priemerný snímok, ktorý bude určovať model pozadia. Tento model ale predpokladá, že je možné získať snímok čistého pozadia a pozadie sa v scéne nemení. Existujú však aj zložitejšie metódy pracujúce s dynamickým pozadím, ktorými sa ale nebudeme zaoberať.

2.4.3 Optický tok

Metóda optického toku zachytáva všetky zmeny za časový interval dt . Každému bodu zodpovedá dvojrozmerný vektor rýchlosti, ktorý opisuje smer a vektor rýchlosti v danom dvojrozmernom obraze. Optický tok teda umožňuje získať informácie o parametroch pohybu objektov v dynamických obrazoch.

2.4.3.1 Výpočet

Pri výpočte optického toku predpokladáme, že hodnota jasu pozorovaných objektov sa v čase nemení a susedné body sa pohybujú podobným spôsobom. Vychádzame z funkcie dynamického obrazu $f(x, y, t)$, ktorá určuje hodnotu jasu

v bode (x, y) v čase t . Pomocou Taylorovho rozvoju a zanedbania vyšších rádov môžeme hodnotu jasv v blízkom okolí vyjadriť vzťahom

$$f(x + dx, y + dy, t + dt) = f(x, y, t) + f_x dx + f_y dy + f_t dt,$$

kde f_x, f_y, f_t označujú parciálne derivácie funkcie f . Pretože platí predpoklad konštantnej jasovej hodnoty bodu

$$f(x + dx, y + dy, t + dt) = f(x, y, t),$$

dostávame

$$-f_t = f_x \frac{dx}{dt} + f_y \frac{dy}{dt}.$$

Cieľom výpočtu optického toku je teda určiť vektor rýchlosti

$$c = \left(\frac{dx}{dt}, \frac{dy}{dt} \right).$$

Všetky pohyby, ktoré môžu nastať, môžeme vyjadriť ako kombináciu štyroch základných.

- translácia kolmá na os pohľadu
- translácia do diaľky
- rotácia okolo osi pohľadu
- rotácia kolmá na osi pohľadu

Tieto druhy pohybov môžeme pomocou analýzy optického toku ľahko rozoznať.

2.4.4 Detekcia významných bodov

Analyzovať pohyb aj v obrazoch snímaných v časových intervaloch, ktoré nemôžeme považovať za dosť malé, môžeme na základe detekcie významných bodov v obraze. Táto metóda pozostáva z dvoch častí:

- nájdenie významných bodov
- určenie korešpondencie medzi týmito bodmi v obraze

Významné body všeobecne označujú miesta čo najmenej podobné svojmu okoliu, napríklad vrcholy alebo hranice objektov.

Pretože k analýze pohybu sú za významné body považované tie, ktorých poloha sa v čase mení, môžeme si pri ich detekcii pomôcť rozdielovou metódou aplikovanou na dané snímky. Medzi najznámejšie detektory vyznamenaných bodov patrí napríklad Moravcov detektor [9] alebo vylepšený Harrisov detektor [8]. Následné hľadanie korešpondencie medzi významnými bodmi všetkých dvoch po sebe idúcich snímok je iteračný proces, ktorého priamou aplikáciou je vytvorenie rýchlostného poľa.

2.5 Technológia MIDI

MIDI, teda **Musical Instrument Digital Interface**, je štandardný komunikačný protokol, ktorý umožňuje hudobným nástrojom, počítačom a iným zariadeniam medzi sebou komunikovať v reálnom čase za účelom prehrávania a nahrávania hudby [23]. Základný dátový blok prenášajúci určitú informáciu sa nazýva „MIDI správa“. Jej špecifikácia je pravdepodobne najdôležitejšou súčasťou MIDI.

V roku 1981 sa na výstave NAMM³ v Anaheimu v USA prejednávali predbežné návrhy na univerzálne rozhranie hudobných nástrojov. O čosi neskôr, 5. 8. 1983, sa objavila finálna verzia MIDI normy 1.0. Štandard bol nejasný a dochádzalo k problémom s kompatibilitou, preto v roku 1984 boli založené orgány zodpovedné za dodržiavanie tejto normy a jej ďalší vývoj - MMA⁴ a JMSC⁵. I v dnešnej dobe je MIDI norma neustále vo vývoji.

Technológia MIDI je síce stará už viac ako 20 rokov, no napriek tomu je stále populárna a užitočná. MIDI je v hudobnom priemysle brané za samozrejmosť a predsa väčšina ľudí na svete o ňom nikdy nepočula, nemajú poňatia, aké možnosti pre tvorbu hudby môže MIDI poskytnúť. Aj naša práca bude využívať pri produkovani hudby MIDI.

MIDI správy

Pravdepodobne najčastejšou správou je **Note On**, obsahuje číslo kanálu, číslo noty a rýchlosť jej prehrania. Pomocou správy **Note Off** je možné vypnutie prehrávania danej noty, obsahuje tiež rýchlosť doznievania. Správa **AfterTouch** je používaná pre prenos informácie o sile stlačenia klávesy. Podobný význam má aj správa **Channel Pressure**, v ktorej je prenášaný priemerný tlak na všetky stlačené klávesy. V správe **Pitch Wheel** sa prenášajú informácie o nastaveniach prvku, ktorým je možné meniť výšku jednotlivých tónov. Pomocou správy **Program Change** je možné meniť typ hudobného nástroja, ktorý sa syntetizér pokúsí napodobiť.

Formát súborov (.mid alebo .smf)

MIDI správy môžu byť uložené do súborov. V štandarde General MIDI je špecifikovaný formát tohoto typu súboru, označovaný tiež skratkou SMF (Standard MIDI File). Ide o binárny súbor, pre ktorý sa používajú najmä prípony *.mid* alebo *.smf*. V MIDI súboroch sú teda uložené správy, doplnené o časovú informáciu, ktoré určujú čas začiatku a konca prehrávania tónu alebo akordu. Súbory sa vďaka jednoduchosti rozšírili nielen medzi hudobníkmi, ale aj medzi bežnými používateľmi osobných počítačov. Malé veľkosti súborov sú taktiež výhodné pre niektoré aplikácie, mobilné telefónne zvonenia, či video-hry. Naša aplikácia bude ukladať skladby práve do tohoto typu súborov.

³National Association of Music Merchants

⁴MIDI Manufacturers Association

⁵Japan MIDI Standard Committee

General MIDI 1

General MIDI, neskôr premenované na General MIDI 1 alebo GM1, nie je to isté ako MIDI. Ak uvažujeme, že MIDI je jazyk, protokol, špecifikácia, typ súboru, General MIDI definuje špecifické vlastnosti MIDI nástrojov. General MIDI zaručuje kompatibilitu, bez neho by napríklad prehrávanie MIDI súborov nebolo konzistentné, zvuk nástrojov by znel úplne inak. Oficiálnu špecifikáciu je možné nájsť v angličtine od MMA. V prílohe tejto práce B sa okrem mien a čísel MIDI nôt nachádza aj zoznam nástrojov GM1.

Kapitola 3

Analýza problému a návrh riešenia

3.1 Podrobný popis problému

Sú dva extrémne pohľady, ktorými sa môžeme na AudioCam dívať. Optimistický pohľad predpokladá riešenie, ktoré umožní používateľovi hudbu ovládať vo svojich rukách, nech už to znamená čokoľvek. V reálnom čase sa podľa jeho predstáv bude hudba vytvárať, človek bude mať pocit, že ju smeruje svojim pohybom. Skeptický pohľad vidí pravý opak, kde užívateľ nezaujato sedí pred webovou kamerou, nie je si istý tým, čo a ako tvorí. Prvý, skôr utopistický pohľad by kvôli nejasnej predstave bránil dokončeniu práce, druhý by nepriniesol nič nové. Tento problém si podrobne popíšeme, aby sme boli schopný navrhnúť priateľský kompromis a určíme si, na čo sa zameriame.

Naším cieľom je vytvoriť virtuálny hudobný nástroj ovládaný pomocou webovej kamery. Predtým, než navrhujeme konkrétne riešenie, sa budeme zaoberať problémami spojenými s návrhom takéhoto nástroja. Hlavným požiadavkom, na ktorý je ale nutné myslieť od začiatku je, že sa o produkovanie hudby budeme snažiť v reálnom čase. Tento požiadavok je rozhodujúci, a preto sa budeme snažiť o minimalizovanie oneskorenia reakcie nášho systému na pohyb užívateľa.

Najskôr sa na problém budeme pozeráť v širšom zmysle. Zabudnime na chvíľu, že našim vstupom je webová kamera a my máme informácie o polohe a pohybe celého ľudského tela. Predstavme si človeka ako rozhranie hudobného nástroja. Pri takejto predstave sú vidieť určité obmedzenia spojené s ľudským telom, pohybom končatín a priestorom, v ktorom sa nachádza. Ako ale takéto rozhranie využiť pri tvorbe hudby? Jednoduchým riešením by bolo priamo spojiť určité pohyby s výslednými zvukmi, podobne ako u klasických akustických nástrojov. Znamenalo by to premeniť parametre pohybu do parametrov zvuku, napríklad celkové zrýchlenie by určovalo tón, veľkosť pohybu by určoval hlasitosť. Malý pohyb hlavy by teda produkoval rôzny výsledok ako pohyb celého tela. Takýchto priamych spojení existuje mnoho, avšak nie je ľahké navrhnúť také, ktoré by najlepšie zohľadňovalo prirodzené požiadavky pri tvorbe hudby.

Na druhej strane by sme nemuseli pohyb mapovať priamo, ale do určitých hudobných parametrov. Vzniká otázka slobody a kontroly takéhoto systému a aké konkrétne vlastnosti dáva pohyb hudbe. Aké teda parametre hudby môžeme ovládať? Na spodnej hranici to sú preddefinované zvuky, tóny alebo sekvencie spus-

titelne danými gestami. Na hornej hranici uvažujeme o robustných systémoch, schopných generovať a ovládať melódiu, tempo, rytmus na základe informácií o pohybe užívateľa. Ak by sme maximalizovali množinu parametrov, pre užívateľa by bolo ťažké získať prehľad a ľahko by sa mohol stratiť, akým spôsobom je hudba vytváraná, pretože viac parametrov vytvára veľkú komplexnosť a zložitosť. Užívateľ by sa tak musel učiť postupne podľa ucha a pamäti, pretože jeho akcie by nemuseli mať okamžitý vplyv na produkovanú hudbu. Hľadalo by sa na ne z celkového kontextu a analyzovali by sa napríklad opakujúce sa pohyby spojené s vytváraním rytmu. Takéto vplyvy užívateľa a jeho pohybov môžu byť ale použité aj menej zrejším spôsobom, podľa zvoleného systému abstrakcie. Dôležité pri týchto mapovaniach je, aby mal užívateľ pocit účasti pri vytváraní skladby, čo priamo závisí od reakcie systému na užívateľa.

Vráťme sa späť k obmedzeniam nášho vstupu. Všetky informácie sa získavajú iba webovou kamerou. Získané snímky musíme v reálnom čase analyzovať a získať z nich informácie o pohybe. Jednou z výhod rozhrania hudobného nástroja založeného na spracovaní obrazu je aj to, že nezaťažuje užívateľa so zariadením samotným. Na druhej strane ho limituje, pretože nemá inú spätnú väzbu ako zvukovú. Tento problém sa čiastočne odstráni rozšírením o kombináciu s vizuálnym výstupom na monitore. Je to však v určitých prípadoch slabá náhrada napríklad v porovnaní s hmatovou odozvou, častou pri hraní na akustické hudobné nástroje. Jednou z úloh teda bude navrhnúť spôsob ovládania, aby aj ľudia, ktorí neboli so systémom oboznámení, sa ho boli schopní naučiť používať. Tento systém bude reagovať na užívateľa a zároveň užívateľ bude reagovať na výstup systému.

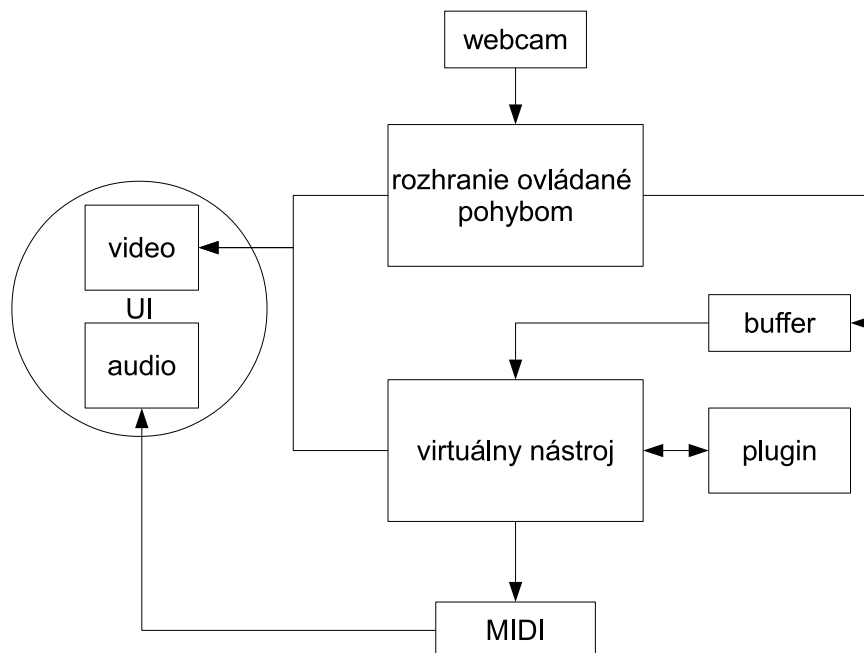
3.2 Dekompozícia problému

Problém rozdelíme na tieto hlavné celky:

- ovládacie rozhranie nástroja zamerané na analýzu pohybu z kamerových snímkov
- virtuálny nástroj produkujúci audio-vizuálny výstup na základe údajov o pohybe
- celkový návrh užívateľského rozhrania

Pri tomto rozdelení vidno, ako všetky celky medzi sebou úzko súvisia. Návrh riešenia jedného je závislý od ostatných. Prvá časť problému, teda vstupné rozhranie ovládané pohybom, bude mať za úlohu kontinuálne spracovávať kamerové snímky a získavať z nich informácie pre ovládanie virtuálneho nástroja. S týmito informáciami ďalej bude pracovať virtuálny nástroj pri tvorbe hudby, preto sa budeme snažiť získať v reálnom čase čo najviac informácií popisujúcich pohyby užívateľa. Toto rozhranie a jeho výstupné parametre by mali byť v určitom smere univerzálne a zameriame na neho najväčšiu pozornosť, aby ho mohlo možné použiť pre rôzne virtuálne nástroje. Tieto informácie budeme vkladať do bufferu, z ktorého si ich virtuálny nástroj bude vyzdvihovať. V práci navrhujeme jednoduchý virtuálny nástroj, ktorý využíva získané dáta o pohybe pri tvorbe hudby a vytvoríme možnosť jednoduchého rozšírenia o plugíny. Pri produkovaní hudby budeme využívať technológiu MIDI. Návrh grafického rozhrania bude primárne

závislý na konkrétnom type virtuálneho nástroja a type získavaných informácií o pohybe užívateľa. Na obrázku 3.1 môžeme názorne vidieť celkové rozdelenie problému.



Obr. 3.1: Schéma návrhu riešenia problému. Snímok z webovej kamery spracuje modul pre pohybové rozhranie. Informácie o pohybe sa ďalej uložia do bufferu, z ktorého si ich vyzdvihne virtuálny nástroj. Ten na ich základe produkuje audio výstup. Grafické užívateľské rozhranie tvoria výstupy z virtuálneho nástroja a pohybového rozhrania.

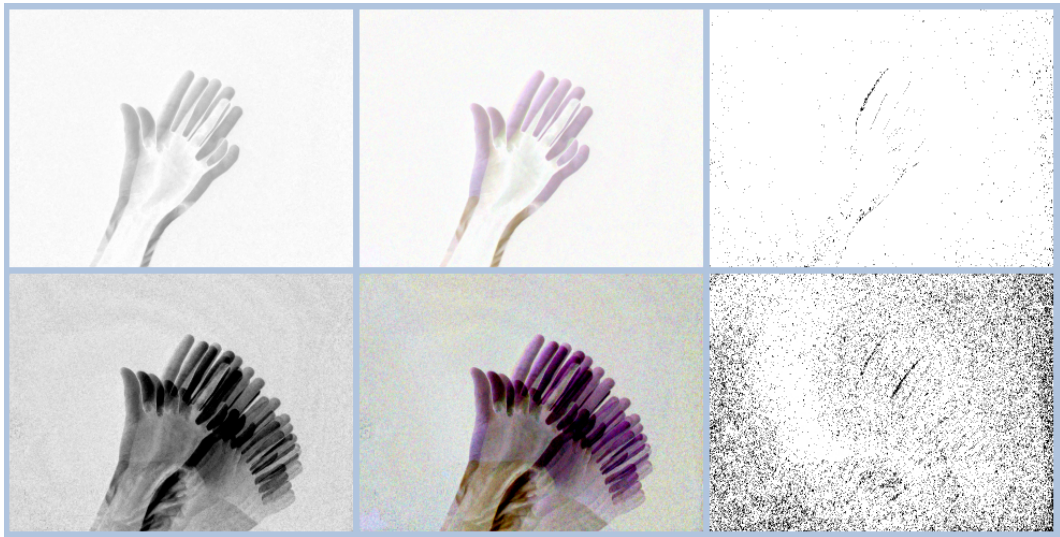
3.3 Ovládanie pomocou pohybu

3.3.1 Aplikácia rozdielovej metódy

Predspracovanie snímkov

V rozdielovej metóde analýzy pohybu budeme pracovať so snímkom v odtieňoch šedej, aj keď máme k dispozícii farebný obraz z webovej kamery. Hlavným dôvodom je to, že v bežných podmienkach je rozdiel medzi použitím farebného obrazu a obrazu v odtieňoch šedej v rozdielovej metóde zanedbateľný. Toto porovnanie môžeme vidieť na obrázku 3.2. Výpočet sa nám takto zrýchli, pretože nemusíme pracovať s tromi farebnými kanálmi, ale iba s jedným.

Ďalej uvažujeme o použitých rozmeroch obrazu. Maximálne rozlíšenie je určené vlastnosťami webovej kamery a je zbytočné škálovať snímok do väčších rozmerov, pretože by sme tým nič nezískali. Uvažujeme preto o zmenšení rozlíšenia snímku s použitím lineárnej interpolácie. Hlavnou výhodou je, že táto operácia je rýchla a nám sa tak zníži celková časová zložitosť výsledného spracovania, pretože pracujeme s menšími snímkami. Na druhej strane, pri nízkom rozlíšení nie sme schopní zachytiť všetky pohyby.



Obr. 3.2: V prvom riadku sa nachádza porovnanie rozdielu pri odčítaní dvoch po sebe idúcich snímkov v odtieňoch šedej (ľavý obrázok) a farebných snímkov (stredný obrázok). Ich rozdiel (obrázok v pravo) reprezentovaný prahovaným binárnym obrazom (zvýraznené pixely s intenzitou väčšou ako nula) z ich absolútneho rozdielu. Je vidieť, že rozdiel pri použití farebného obrazu a obrazu v odtieňoch šedej pri aplikácii rozdielovej metódy je nepatrný šum obrazu a nemá vplyv na výsledne spracovanie. V druhom riadku pre porovnanie a zvýraznenie sčítaných 10 po sebe idúcimi rozdielových obrazov a ich absolútnych rozdieloch.

V každom snímku sa nachádza aj určitá miera šumu, ktorú sa budeme snažiť potlačiť. Použijeme filter Gaussovského rozostrenia. Opäť si musíme dať pozor pri veľkosti použitej matice, pretože sa snímok zároveň rozmaže.

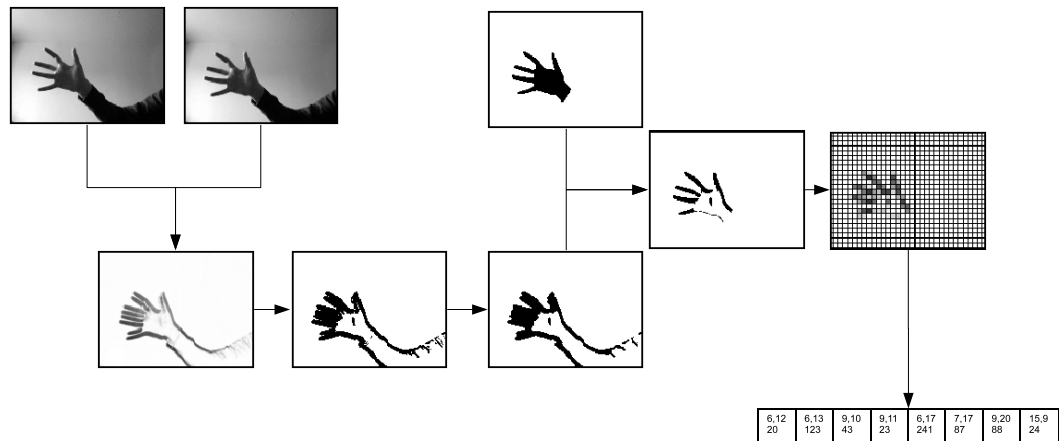
Rýchle odstránenie pohybov nepodstatných častí tela

Niekedy by sme chceli ovládať virtuálny nástroj iba rukami a zvyšné pohyby tela by sme chceli ignorovať. Prepokladajme, že farba pokožky, ani jej odtiene, nie sú zastúpené nikde inde v snímku. Pomocou spodnej a hornej hranice farby odtieňu pokožky, napríklad vo farebnom modele RGB, by sme sa mohli jednoducho zbaviť pohybov iných častí užívateľovho tela alebo objektov v pozadí. Vytvorili by sme masku rúk z aktuálneho snímku, ktorá by obsahovala iba body, ktorých farba by padla do tohoto rozsahu. Nechali by sme na užívateľovi, ktorý by podľa daných podmienok rozhodol, či použije tento spôsob a zároveň by ručne nastavil hornú a dolnú hranicu, v ktorej spadajú odtiene pokožky jeho ruky.

Spracovanie snímkov

Na určenie miest, kde nastal pohyb, sa použije rozdielová metóda popísaná v 2.4.1 s použitím dvoch po sebe idúcich predspracovaných snímkov, alebo viacerých tzv. metódou kumulovaného obrazu. Jej výsledkom je binárny obraz, ktorý popisuje detekované miesta pohybu. Naň sa aplikuje filter medián, ktorým odstrániť lokálny šum. Ak je k dispozícii maska rúk, odfiltrujú sa pohyby ostatných častí tela. Dôležité pri tomto postupe je správne určenie prahu pri rozdielovej metóde a veľkosti matice použitej pri aplikovaní filtra medián.

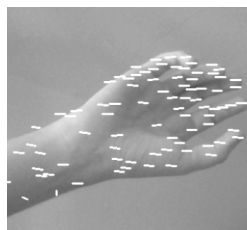
Následne sa výsledný binárny obraz rozdelí na $n \times m$ oblastí. Z tých buniek v tomto rozdelení, v ktorých nastal nejaký pohyb, sa vytvorí zoznam obsahujúci informácie o súradniciach a počte bodov pohybu. Tento zoznam bude tvoriť základ informácií pre ďalšie spracovanie virtuálnym nástrojom. Kroky výpočtu sú znázornené na obrázku 3.3



Obr. 3.3: Názorné zobrazenie postupu získania informácií o pohybe rozdielovou metódou. Najskôr sa určí absolútny rozdiel dvoch po sebe idúcich predspracovaných snímok a potom sa podľa zvoleného prahu vytvorí rozdielový obraz. Pomocou filtra medián sa odstráni lokálny šum. Ďalej sa tento binárny obraz prekryje s maskou, ktorá vyberá iba body vo zvolenom farebnom rozsahu v pôvodnom snímku. Ako výstup sa určí zoznam oblastí s nenulovým počtom bodov pohybu.

3.3.2 Optický tok

Okrem informácie, kde nastal pohyb, chceme získať aj vektor rýchlosti pohybu, optický tok. Lucas a Kanade [14] navrhli algoritmus na určenie optického toku v celom obraze, ktorý je základom pre KLT(Kanade-Lucas-Tomasi) algoritmus [15] na sledovanie významných bodov. Spôsob hľadania významných bodov navrhujú vo svojej práci Shi a Thomaši [13]. My optický tok určíme iba pre významné body v miestach pohybu daného snímku a to iba v prípade, ak bol zaznamenaný pohyb pomocou rozdielovej metódy. Na hľadanie významných bodov a ich sledovanie využijeme optimalizované metódy knižnice OpenCV. Výsledok môžeme vidieť na obrázku 3.4



Obr. 3.4: Znázornenie optického toku pri posune ruky doprava.

3.4 Virtuálny nástroj

Úlohou virtuálneho nástroja je použiť získané informácie o pohybe pri vytváraní hudby. Budeme produkovať tóny s použitím technológie MIDI, kde budeme najmä posielat správy Tone On na začatie prehrávania tónu a Tone Off na jeho vypnutie. Užívateľovi je k dispozícii 128 nástrojov, ktorých zoznam môžeme nájsť v prílohe práce B. AudioCam sme sa rozhodli navrhnuť tak, aby ho bolo možné jednoducho rozšíriť o nové virtuálne nástroje pomocou pluginov. Tieto tak môžu rôznym spôsobom pristupovať k produkovaniu hudby. Aby sme ale demonštrovali využitie rozhrania ovládaného pohybom pri tvorbe hudby, navrhli sme jednoduchý virtuálny nástroj, ktorý simuluje hranie na klávesoch. Užívateľ na obrazovke vidí klávesy a na tento nástroj sa hrá jednoduchým spôsobom. Ak bol detekovaný pohyb v oblasti klávesu, zahrá sa konkrétny tón, pričom sa zároveň vytvorí vizuálny efekt. Dôležité pri tomto nástroji je navrhnuť rozloženie klávesov, pretože pri hustom rozložení užívateľ bude hrať nechtiac aj to, čo nechce. Preto sme sa rozhodli mať iba osem kláves stupnice, ktoré sú rovnomerne rozložené vedľa seba, viď obrázok 3.3.

Kládli sme väčší dôraz na rozhranie, ktoré získava informácie o pohybe, a preto je náš návrh nástroja jednoduchý. Aby sme ale boli schopní v budúcnosti rozšíriť možnosti jeho využitia, vytvorili sme priestor pre pridávanie pluginov. Pomocou nich je tak ľahké neskôr pridať rôzne spôsoby tvorenia hudby reagujúcej na pohyb.

Kapitola 4

Implementácia

4.1 Použité technológie

4.1.1 Programové prostredie

Program je napísaný v objektovo orientovanom jazyku C# na platforme .NET. Projekt je vytvorený a skompilovaný pomocou vývojového prostredia Microsoft Visual Studio 2010.

4.1.2 Podporné knižnice

- **EmguCV** [24] je multiplatformový .NET obal (wrapper) knižnice na spracovanie obrazu **OpenCV**[25]. Dovoľuje volanie a použitie jej funkcií z jazyka C#. Inštalácia EmguCV je dostupná z:
<http://sourceforge.net/projects/emgucv/files/latest/download>.
- **C# MIDI Toolkit** je knižnica umožňujúca používať v projekte MIDI. Projekt je dostupný z:
<http://www.codeproject.com/Articles/6228/C-MIDI-Toolkit>.
- **directshow.net** umožňuje prístup k funkcionalite Microsoft DirectShow zvnútra .NET aplikácie. My ju využívame pri získaní informácií o webovej kamere. Domovská stránka a zdrojové kódy projektu projektu:
<http://directshownet.sourceforge.net/>.

4.1.3 Zostavenie projektu

Do referencií projektu aplikácie sú pridané knižnice pre spracovanie obrazu EmguCV (Emgu.CV.dll, Emgu.Util.dll a Emgu.CV.UI.dll), knižnice pre používanie MIDI (Sanford.Collections.dll, Sanford.Multimedia.dll, Sanford.Multimedia.Midi.dll, Sanford.Multimedia.Timers.dll, Sanford.Threading.dll) a knižnice directshow.net (DirectShowLib-2005.dll). Do projektu sú pridané taktiež knižnice OpenCV.

4.2 Štruktúra projektu

AudioCam je rozdelený do troch projektov:

- **AudioCamLib**
knižnica, ktorá obsahuje spoločné triedy použité v aplikácii a v pluginoch
- **AudioCam**
spustiteľná aplikácia
- **AudioCamPlugin**
príklad vytvorenia pluginu virtuálneho nástroja

4.3 Dekompozícia úlohy

Na implementačnej úrovni je projekt rozdelený na tieto celky:

- **Analýza pohybu**
spracovanie snímaných obrazov a detekcia pohybu
- **Virtuálny nástroj**
spracovanie informácií o pohybe do podoby tónov
- **Rozšíriteľnosť**
pridávanie nových pluginov
- **Grafický výstup a užívateľské rozhranie**
okná aplikácie
- **Kalibrácia a nastavenia**

Opíšeme zjednodušený postup kroku riešenia problému. Po získaní každého snímku z webovej kamery sa určia aktuálne informácie o vykonávaných pohyboch. Tieto informácie sa uložia a virtuálny nástroj si ich následne vyzdvihne. Na ich základe produkuje tóny, ktoré určujú vytváranie efektov.

4.4 Popis tried a objektov programu

Hlavný objekt je triedy *AudioCam*, ktorého metóda *ProcessFrame()* zabezpečuje kontinuálny priebeh spracovania vstupu. Volá sa opakovane vždy po každom spracovaní snímku a vykonáva sa v nej všetko potrebné.

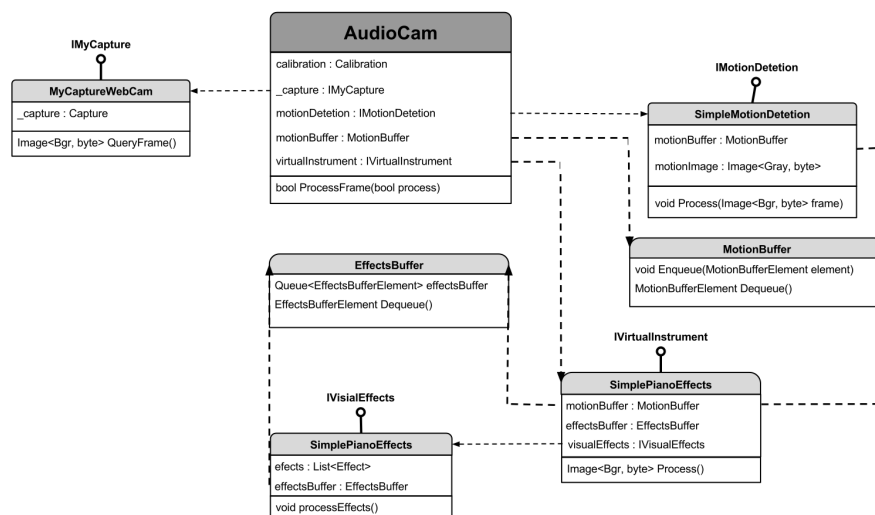
Najskôr sa vyzdvihne snímok z webovej kamery, pomocou metódy objektu *_capture*. Tento objekt implementuje interface *IMyCapture*.

Nový snímok z webovej kamery sa predá ďalej cez interface *IMotionDetetion* objektu *motionDetetion*, ktorý zabezpečuje spracovanie pohybu. Výsledkom sú informácie o pohybe pre rozhranie virtuálneho nástroja, ktoré uloží do *motion-Bufferu*.

Z *motionBufferu* si následne tieto informácie o pohybe vyberie objekt *virtualInstrument* a zahrá tóny (napríklad odpovedajúce stlačeným klávesám). Trieda každého virtuálneho nástroja musí implementovať interface *IVirtualInstrument*. V základnom móde sa podľa zahratých tónov vytvoria informácie pre efekty, ktoré sa vložia do *effectBufferu*.

Tieto efekty sa spracujú objektom *visualEffects*. Jeho trieda implementuje interface *IVisualEffects*. Konkrétne v móde klavíru je objekt pre efekty triedy *SimplePianoEffects*.

Objekt *audioCam* potom podľa nastavení zloží výsledný obraz výstupu. Ten sa môže skladať z pôvodného snímku, zo zvýraznených častí pohybu a pridaných efektov. Všetky nastavenia sa nachádzajú v objekte triedy *Calibration* ako hodnoty jeho premenných. Schéma hlavných tried projektu je znázornená na obrázku 4.1.



Obr. 4.1: Prehľad a rozdelenie hlavných tried a rozhraní programu

Získanie snímkov z webovej kamery

Pre vyzdvihnutie snímku z webovej kamery sa využíva objekt triedy *Capture* z knižnice **EmguCV**, ktorý všeobecne slúži na získavanie snímkov z kamery ako aj z videa.

Tento objekt sme v projekte obalili vlastným rozhraním *IMyCapture*.

Analýza pohybu

Pri spracovaní obrazu sa využívajú volania metód knižnice *OpenCV* opäť prostredníctvom *EmguCV*. Spracovanie pohybu má na starosti trieda *SimpleMotionDetetion*.

Na komunikáciu s objektom využívame interface *IMotionDetetion*.

Predspracovanie snímku

Predtým, ako sa zistí pohyb, snímky z webovej kamery sa upravujú podľa aktuálnych nastavení. Prvou úpravou snímku je zmena veľkosti. Toto má na starosti metóda *preprocessframe()*. Ak je nasnímané pozadie bez užívateľa, môže sa vyrábať maska, určujúca pozadia a popredie.

Následne sa podľa nastavení môže pracovať iba s určitými časťami snímky, ktorých farby bodov sú v určitom rozsahu, napríklad pre odfiltrovanie nepodstatných, alebo rušivých pohybov patriacich iným častiam tela a zameranie sa na pohyb rúk, ktorých farba sa pohybuje v určitom rozsahu. Toto implementuje metóda *calculateInRange()*.

Detekcia pohybu

V metóde *calculateMotionImage()* sa na detekciu pohybu použije diferenciálna metóda analýzy pohybu s možnosťou použitia kumulovaného obrazu. Získame tak informácie o zmene jasů v každom bode. Na odstránenie malých lokálnych zmien alebo šumu používame medián filter.

Optický tok

V metóde *opticalFlow()* určíme optický tok. Najskôr sa nájdu významné body pomocou metódy *GoodFeaturesToTrack()*. Potom použitím Lucas-Kanade pyramidovej metódy, funkcia *OpticalFlow.PyrLK()*, sa vyráta optický tok.

Spracovanie pohybu

Samotná informácia o zmene ešte nestačí, ďalej ju spracujeme pre konkrétne použitie. Vytvoríme z nej informácie pre ovládacie rozhrania nástroja, ktoré vložíme do bufferu, kde si ich neskôr vyzdvihne virtuálny nástroj, ktorý pomocou nich vytvára hudbu. Do bufferu ukladáme informácie o pohyboch pre daný snímok. Metóda, ktorá transformuje informácie o pohybe do rozhrania nástroja, je *calculateMotionTones()*.

Virtuálny hudobný nástroj

Produkovanie hudby má na starosti objekt *virtualInstrument*, môže byť triedy *PianoVirtualInstrument*, *PluginExtentionVirtualInstrument* alebo *FreeModeVirtualInstrument*. Typ objektu určuje mód virtuálneho nástroja.

Virtuálny nástroj si najskôr vyzdvihne z *motionBufferu* aktuálne pohyby. Tie prejde a podľa nich určí odpovedajúce tóny. Pri zahrnutí tónov sa následne uložia informácie do *effectBufferu* pre spracovanie efektov.

Pluginy

Virtuálny nástroj triedy *PluginExtentionVirtualInstrument* je rozšíriteľný o pluginy. Trieda *MyPlugins* drží zoznam aktuálnych načítaných pluginov a aktuálny používaný plugin. Pomocou *PluginManagera* sa načítavajú pluginy z *.dll* súborov, konkrétne triedy implementujúce *IACPlugin*. *PluginManager* využíva objekt triedy *RemoteLoader*, ktorý sa stará o načítavanie pluginov a beží v aplikačnej doméne *pluginManagera*. Komunikáciu medzi aplikačnými doménami zabezpečuje dedenie od *MarshalByRefObject*. Plugin teda musí implementovať interface *IACPlugin*, ktorého hlavná metóda je *Execute(IACPluginContext pcontext)*. Volá sa pri spracovaní *motionBuffera*. Parameter metódy implementuje interface *IACPluginContext*. Plugin zistí informácie o zachytenom pohybe (*motionBufferElement*) a odošle nasledujúce tóny (*NewTones*).

Efekty

Efekty sa tvoria pri vydaní tónu. Objekt `visualEffect` implementuje interface `IVisualEffect`, konkrétne trieda `SimplePianoEffect`, si vyzdvihne z `effectBuffera` informácie o tónoch a vytvorí jednoduché pohybujúce sa kruhové efekty. Klávesnica klavíru sa vykresľuje v metóde `drawKeyBoard()`.

Nastavenie a kalibrácia

Všetky dôležité nastavenia drží objekt triedy *calibration*. Ide o nastavenie pre zachytenie pohybu (pre veľkosť snímku). Veľkosť hodnoty parametrov pre rozmazanie, thresholding, medián filter, pre virtuálny nástroj, pre zobrazovanie efektov atď). Môžu sa uložiť alebo načítať pomocou XML serializéra.

Užívateľské rozhranie

Hlavné okno je triedy *AudioCamMainForm*. Cez toto okno sa otvárajú ostatné. Tu sa inicializuje objekt *audioCam*. V tomto okne sa môže vypnúť/zapnúť webová kamera. Podľa jej stavu sú potom aktívne tlačidlá Play a Nastavenie.

Okno triedy *CalibrationForm* slúži na nastavenie hodnôt určených pre kalibráciu. Dokáže nastavenia uložiť, otvoriť alebo reštartovať do pôvodných. V metóde *processCalibration()* nastavuje výstupné obrazy.

V okne triedy *PlayKeyboardForm* hrá užívateľ na samotný virtuálny nástroj. Dôležitou metódou je *processScene()*, v ktorej sa volá metóda objektu *audioCam* *processFrame()*. Ďalej obsahuje metódy pre nahrávanie a uloženie hudby.

Kapitola 5

Užívateľská dokumentácia

5.1 Inštalácia a spustenie programu

Minimálne požiadavky

Program beží na operačnom systéme Windows 7, minimálne s procesorom 1GHz, operačnou pamäťou 1GB a **pripojenou webovou kamerou**.

Inštalácia programu

Program sa nainštaluje spustením súboru *setup.exe*. Program sa odinštaluje pomocou položky v Control Panely Add or remove programs (alebo Pridať alebo odobrať programy).

Spustenie programu

Program sa spustí pomocou súboru *audiocam.exe*, ktorý sa nachádza v adresári aplikácie.

5.2 Ovládanie programu a popis užívateľského rozhrania

Po spustení sa zobrazí hlavná obrazovka, na ktorej sa nachádza hlavné tlačidlo Play a štyri vedľajšie postranné tlačidlá (menované v protismere hodinových ručičiek):

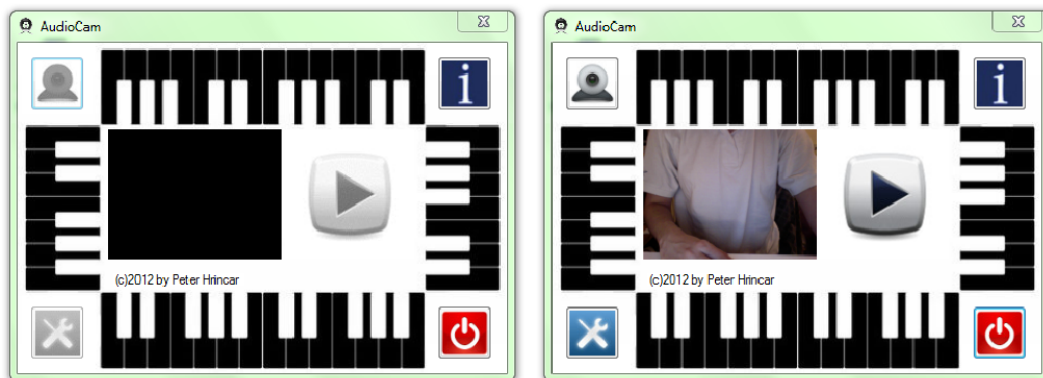
- zapnutie/vypnutie webovej kamery
- nastavenia a kalibrácia programu
- ukončenie programu
- informácie o programe

Tlačidlo webovej kamery môže mať tri stavy v závislosti od jej fungovania:

- **aktívny**
program používa kameru, po stlačení ju vypne

- **neaktívny**
program nepoužíva kameru, po stlačení sa ju pokúsi zapnúť
- **chybový**
webová kamera je nedostupná, buď to nie je pripojená, alebo nastal iný problém; po jeho stlačení sa ju program pokúsi opätovne zapnúť

Tlačidlá *Play* a *nastavenia* sú aktívne iba pri zapnutej kamere. Okrem toho sa v hlavnom okne nachádza náhľad aktuálnych snímkov.



Obr. 5.1: Módy hlavného okna aplikácie AudioCam, naľavo s vypnutou webovou kamerou, neaktívne tlačidlá *Play* a *nastavenia*, napravo so zapnutou kamerou, s aktívnymi všetkými tlačidlami a náhľadom

Virtuálny nástroj

Po stlačení tlačidla *Play* sa zobrazí okno, v ktorom užívateľ ovláda virtuálny hudobný nástroj pomocou svojho pohybu zachyteného webovou kamerou.

Vyprodukovanú hudbu si môže nahráť a následne uložiť do súboru vo formáte *mid*. V tomto okne má možnosť meniť základné nastavenia:

- zobrazovať snímky z webovej kamery
- zapnúť/vypnúť zvuk
- zobrazovať pohyby
- zobrazovať efekty
- meniť použitý nástroj
- prepínať mód virtuálneho nástroja

Nahrávanie a ukladanie hudby

Po stlačení tlačidla **REC** sa začne nahrávať, až kým sa stlačí **STOP**. V nahrávacej fáze nie je možné prepnúť mód virtuálneho nástroja. Po nahratí skladby sa zviditeľní tlačidlo **Save**, ktoré umožňuje nahrávku uložiť do súboru vo formáte *.mid*.

Módy

Módy nástroja sa menia stlačením tlačidla **Change Mode**.

Základný mód

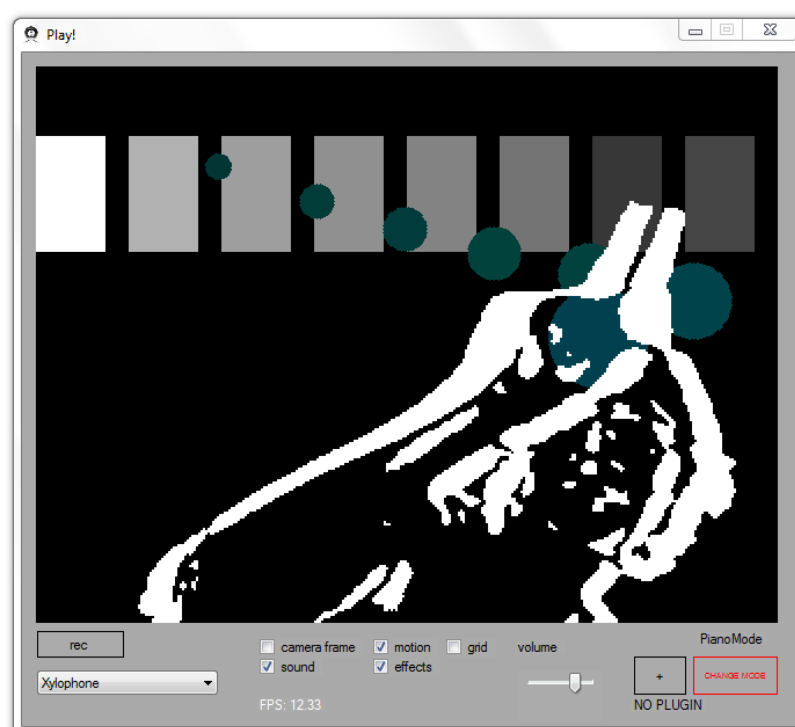
Základný mód virtuálneho nástroja s klávesnicou vydá tón podľa toho, či v oblasti klávesu nastal pohyb. V nastaveniach aplikácie si užívateľ môže vybrať, na aký hudobný nástroj hrá.

Free mód

V tomto móde užívateľ pomocou smeru a rýchlosti pohybu ovláda loptičku, ktorá pri odraze od stien vydá tón určený pozíciou jej nárazu. Tento mód je experimentálny a slúži na demonštráciu čiastočného využitia optického toku.

Rozšíriteľný mód

Užívateľ môže v tomto móde nahráť plugin, ktorý určí vlastnosti virtuálneho nástroja. Cez tlačidlo **+** sa otvorí nové okno, kde sa vyberie plugin. V tomto okne sa pomocou tlačidla **Import plugins from dir.** importujú súbory (*dll*) zvoleného adresára obsahujúce pluginy.

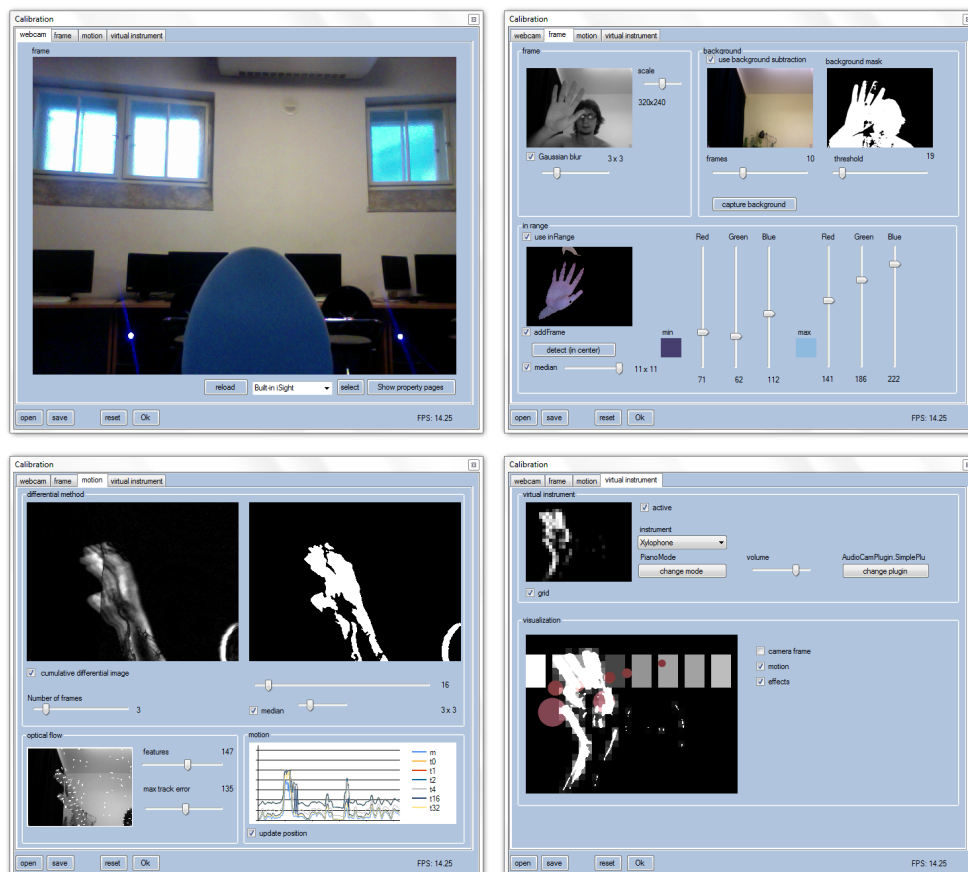


Obr. 5.2: Okno Play programu AudioCam.

5.3 Nastavenia, kalibrácia programu

Cez hlavné okno aplikácie sa užívateľ môže pri zapnutej webovej kamere dostať do sekcie **Nastavenia a kalibrácie programu**. Toto okno je rozdelené na sekcie podľa zamerania:

- **webcam**
táto sekcia zobrazuje aktuálne snímky
- **frame**
možnosť nastavenia predspracovania snímkov
- **motion**
nastavenia a kalibrácie detekcie pohybu
- **virtual instrument**
nastavenie virtuálneho nástroja



Obr. 5.3: Možnosti okna *calibration* programu AudioCam.

Frame

- *frame*

Nastavenie rozlíšenia spracovaného snímku, jeho znížením sa všeobecne zvyšuje celková rýchlosť spracovania.

Možnosť vyhladenia snímku a odstránenia šumu pomocou Gaussovského rozostrenia, plus nastavenie veľkosti konvolučného jadra pri jeho použití.

- *background*

Po stlačení tlačidla *Capture frame* užívateľ musí odstúpiť zo scény a po čase troch sekúnd sa nasníma pozadie. Pozadie je spriemerované z n snímok, ktorých počet sa dá nastaviť. V tejto časti je taktiež zobrazená maska, určujúca objekty a pozadie (binárny obrázok, čierna je pozadie, biela popredie). Odrátanie pozadia sa môže potencionálne využívať v budúcnosti, v základnom móde virtuálneho nástroja sa však nepoužíva (ide skôr o demonštratívne zobrazenie).

- *in range*

Užívateľ môže určiť oblasti, z ktorých budú spracovávané informácie o pohybe. V týchto oblastiach budú iba body, ktorých farba padne do zvoleného rozsahu. Rozsah sa volí vo farebnom modele RGB. Určí sa minimálna a maximálna farba. Pomocou tlačidla *Detect* sa použije spriemerovaná farba z oblasti stredu snímku, preto by užívateľ mal pri jeho stlačení mať ruku v dostatočne blízkej vzdialenosti a v strede snímku. Získaný rozsah je len orientačný a užívateľ ho musí doladiť ručne. Pomocou prepínača môže použiť *medián* filter na odstránenie lokálnych šumov a pomocou *addFrame* zobrazí snímok s použitím výslednej masky. Toto nastavenia sa však dá využiť iba pri dobrých vonkajších podmienkach, inde v obmedzenej miere, preto je primárne vypnuté.

Motion

- **differential method**

Rozdielová metóda analýzy pohybu môže byť založená na odčítaní dvoch posledných snímok alebo na použití kumulovaného rozdielového obrazu. Pri kumulovanom obraze užívateľ určí počet posledných snímok, z ktorých sa vytvorí výsledok. Veľké hodnoty budú mať za následok dojazdový efekt, ktorý nepriaznivo ovplyvní výsledok.

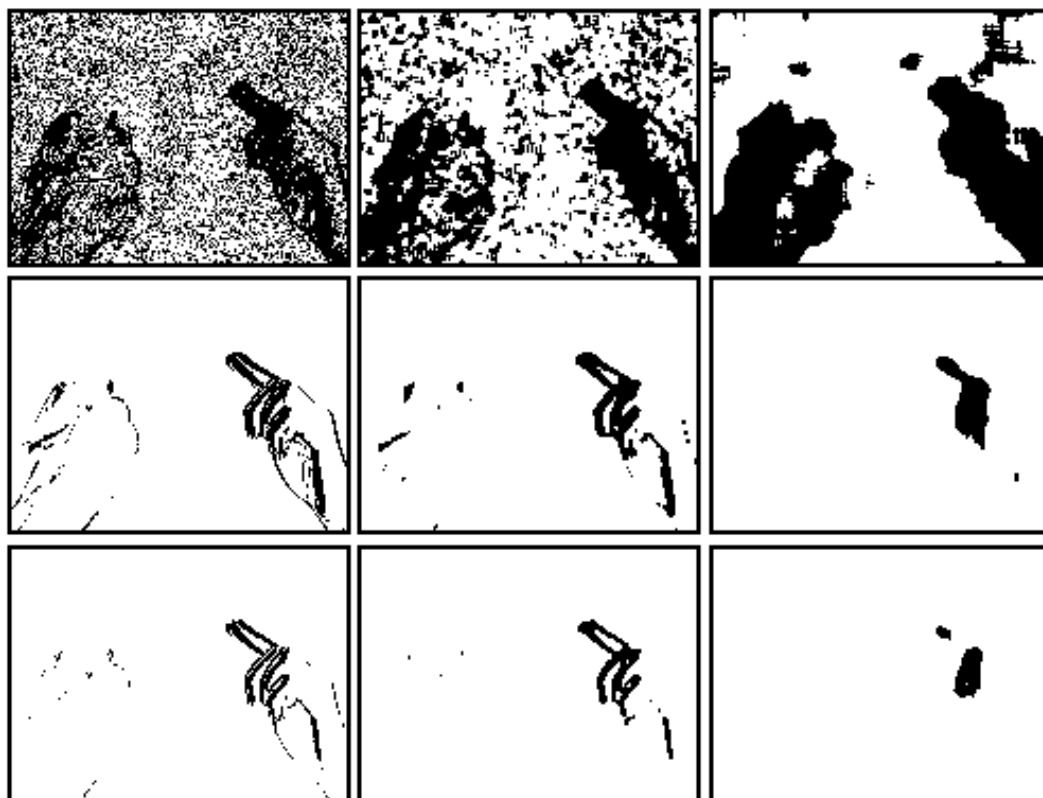
Z rozdielového, respektíve kumulovaného obrazu sa určia miesta zachyteného pohybu na základe určeného prahu. Nastavenie prahu (*threshold*) je v rozsahu 0 - 255. Príliš nízky prah bude spôsobovať zachytávanie šumu, príliš vysoký bude filtrovať malé zmeny. Na odstránenie lokálneho šumu sa môže aplikovať *medián* filter, užívateľ určí veľkosť použitého jadra.

- **optical flow**

Užívateľ určí počet významných bodov, ktoré sa budú sledovať. Z týchto bodov sa určí optický tok. Príliš nízky počet bude mať za následok chyby a príliš vysoký bude spomaľovať rýchlosť spracovania pohybu, preto je nutné spraviť kompromis. Významné body sa určujú iba v miestach pohybu označených rozdielovou metódou.

- **motion**

Orientačný graf celkového množstva pohybu s rôznym nastavením veľkosti prahu.



Obr. 5.4: Vplyv základných parametrov na spracovanie pohybu. Zhora dole sa zvyšuje veľkosť použitého prahu, sprava doľava sa zvyšuje použitá hodnota matice filtra medián. Pri nízkom nastavení prahu sa namiesto pohybu zachytáva šum, pri vysokom sa veľká časť pohybu ignoruje. Pri malej matici veľkosti filtra medián sa filtruje iba šum, pri veľkej aj malé pohyby.

Virtual instrument

V tejto sekcii sú časti venované virtuálnemu nástroju a výstupu programu:

- **virtual instrument**

nastavenia virtuálneho nástroja.

- **vypnutie/zapnutie** priebehu spracovania pomocou prepínača *active*,
- **nastavenie hlasitosti**,
- **výber nástroja** použitého v základnom móde,
- **zmena módu nástroja**, postup rovnaký ako v hlavnom okne *Play*,
- **zmena pluginu**, postup taktiež rovnaký ako v okne *Play*.

- **visualization**

Nastavenia grafického výstupu. Zobrazovanie snímaných snímkov (*camera frame*), zachytených bodov (*motion*) a oblastí zachyteného pohybu (*grid*), efektov (*effects*). Podobne ako tie nachádzajúce sa v hlavnom okne *Play*.

5.4 Riešenie problémov

Môže nastať situácia, kedy je tlačidlo webovej kamery aktívne, ale kamera nefunguje správne. Buď sa program pokúsil pripojiť a webová kamera je už používaná, obsadená, iným programom (žiaden obraz), alebo užívateľ kameru odpojil (striedajúce sa posledné prevrátené snímky). V takom prípade musí užívateľ chybu odstrániť (opätovne zapnúť webovú kameru, alebo zavrieť program, ktorý ju používa) a následne ju zapnúť a vypnúť pomocou tlačidla webovej kamery v programe.

Záver

Naším cieľom bolo navrhnutie a implementovanie programu, prostredníctvom ktorého by užívateľ pomocou webovej kamery skladal hudbu svojim pohybom. Tento cieľ sa nám podarilo naplniť. Zamerali sme sa najmä na získavanie informácií o pohybe v reálnom čase. Navrhli sme univerzálne rozhranie virtuálneho nástroja. Informácie o pohybe získavame pomocou rozdielovej metódy analýzy pohybu a optického toku. Ako aplikáciu sme vytvorili simuláciu kláves, na ktorých užívateľ hrá. Hudbu sme produkovali pomocou technológie MIDI. Užívateľ má možnosť si svoju vytvorenú hudbu nahráť a uložiť do súboru. Prínos našej práce vidíme hlavne v tom, že sme celkové riešenie navrhli tak, aby ho bolo možné v budúcnosti jednoducho rozširovať.

V kapitole 1 sme sa venovali súčasnému stavu poznania v tejto oblasti. Popísali sme históriu a vývoj nástrojov založených na pohybe. Porovnávali sme práce podobného zamerania. Ďalej v kapitole 2 predkladáme základné teoretické podklady, ktoré boli potrebné pri návrhu riešenia problému. Dôležitá pre nás bola hlavne rozdielová metóda analýzy pohybu a optický tok. Popísali sme technológiu MIDI, ktorú sme použili pri produkovani hudby. V kapitole 3 sme analyzovali problém najskôr zo širšieho pohľadu a potom s ohľadom na naše obmedzenia — získavanie dát z webovej kamery. Rozdelili sme riešenia problému na tri časti — ovládanie pohybom, virtuálny nástroj a užívateľské rozhranie. Našu pozornosť sme zamerali na časť rozhrania virtuálneho nástroja ovládaného pohybom. Popísali sme získavanie informácií o pohybe s použitím rozdielovej metódy detekcie pohybu. Optický tok sme sa rozhodli určovať iba pre významné body v obraze. Využitie sme demonštrovali jednoduchým virtuálnym nástrojom, ktorý simuloval hranie na klávesy. Návrh časti nástroja však počítal s neskorším použitím pridávania pluginov. V kapitole 4 o implementácii sme bližšie popísali projekt z programátorského hľadiska. V poslednej kapitole 5 sa venujeme používaniu programu z pohľadu užívateľa.

Pri vývoji aplikácie bolo výhodné mať možnosť kalibrovať všetky dôležité parametre, a tak docieľiť optimálne nastavenia fungovania programu. Na druhej strane musíme byť v tomto bode kritickí, pretože pre priemerného užívateľa je výhodnejšia automatická kalibrácia, ktorá v programe chýba. Užívateľ nemusí rozumieť významu parametrov a môže ich nastaviť tak, že sa program stane nepoužiteľný. Napríklad sa zníži odozva, alebo sa pohyb prestane správnym spôsobom detektovať.

Pri pohľade do budúcnosti sú možnosti ďalšieho rozšírenia pomerne široké. Chceli by sme viac rozvíjať získavanie informácií o pohybe zo snímok kamery. Ďalej vidíme veľký potenciál aplikácie v pokračovaní návrhu spôsobu interaktívneho vytvárania hudby, ktorá je ovládaná pohybom. V tomto smere by sme chceli aj naďalej pokračovať a zapojiť do tvorivého procesu hudobných skladateľov.

Zoznam použitej literatúry

- [1] SONKA, Milan a Vaclav, HLAVAC a Roger, BOYLE. *Image Processing, Analysis and Machine Vision*. 2. vydanie. Pacific Grove: PWS Publishing, 1999. ISBN 0-534-95393-X.
- [2] Vaclav, HLAVAC a SONKA, Milan. *Počítačové vidění*. Grada 1992 ISBN 80-85424-67-3.
- [3] JAIN, Ramesh a Rangachar, KASTURI a Brian, SCHUNCK. *Machine Vision*. 1. vydanie. New York: McGraw-Hill, 1995. ISBN 0-07-032018-7.
- [4] RAFAEL, C. Gonzalez a Richard E., WOODS. *Digital Image Processing*. 2. vydanie. Pacific Grove: PWS Publishing, 2007. ISBN 0-13-168728-X.
- [5] FELS, Sidney S. a Kenji, MASE. *Iamascope: A Graphical Musical Instrument..* Computers and Graphics. Volume 2. No. 23. 1999.
- [6] BERG, Tamara, et al. *Interactive Music: Human Motion Initiated Music Generation Using Skeletal Tracking By Kinect*.
- [7] WOLFRAM, Stephen. *A new kind of science*. 1. vydanie. Champaign : Wolfram Media, 2002 ISBN 1-57955-008-8.
- [8] HARRIS, C. and M. STEPHEN. *A combined corner and edge detection*. In Proceedings of the 4th ALVEY vision conference, pages 147–151, University of Manchester, England, September 1988.
- [9] MORAVEC, H. *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*. Tech Report CMU-RI-TR-3, Carnegie-Mellon University, Robotics Institute, September 1980.
- [10] JEDLIČKA, Richard. *Ovládání počítače pomocí jednobarevných objektů snímáných kamerou*. Bakalářská práce, Univerzita Karlova v Praze, Matematicko-fyzikální fakulta, 2012.
- [11] LAPIN, Peter. *Minority Report*. Bakalářská práce, Univerzita Karlova v Praze, Matematicko-fyzikální fakulta, 2008.
- [12] ŠEDIVÝ, Miroslav. *Ovládání PC pomocí webové kamery*. Bakalářská práce, Univerzita Karlova v Praze, Matematicko-fyzikální fakulta, 2009.
- [13] SHI, Jianbo a Carlo TOMASI “Good features to track” in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’94), (Seattle), s. 593–600, IEEE Computer Society, June 1994.

- [14] LUCAS, D. Bruce a Takeo KANADE *An Iterative Image Registration Technique with an Application to Stereo Vision* International Joint Conference on Artificial Intelligence, s. 674-679, 1981.
- [15] TOMASI, Carlo a Takeo KANADE *Detection and Tracking of Point Features*. Carnegie Mellon University Technical Report CMU-CS-91-132, April 1991.
- [16] THEREMIN WORLD. *Theremin*
<http://www.thereminworld.com/Article/14232/what-s-a-theremin-/>
 [ONLINE] (3/8/12)
- [17] MEDIA ART NET. *John Cage Variations V*
<http://movimientolaredsd.ning.com/video/john-cage-variations-v>
 [ONLINE] (3/8/12)
- [18] NIME. *New Interfaces for Musical Expression*
<http://www.nime.org/> [ONLINE] (3/8/12)
- [19] INTERNATIONAL COMPUTER MUSIC ASSOCIATION. *International Computer Music Conference*
<http://www.computermusic.org/page/23/> [ONLINE] (3/8/12)
- [20] WOLFRAM RESEARCH. *How WolframTones Works*
<http://tones.wolfram.com/about/how.html> [ONLINE] (3/8/12)
- [21] CAM-TRAX TECHNOLOGIES. *CamSpace*
<http://www.camspace.com> [ONLINE] (3/8/12)
- [22] COMPUTER LABORATORY, UNIVERSITY OF CAMBRIDGE
Prvá webová kamera
<http://www.cl.cam.ac.uk/coffee/qsf/coffee.html> [ONLINE] (3/8/12)
- [23] *Oficiálne stránky venované MIDI.*
 MIDI MANUFACTURERS ASSOCIATION INCORPORATED
<http://www.midi.org/> [ONLINE] (3/8/12)
- [24] *EmguCV*
http://www.emgu.com/wiki/index.php/Main_Page [ONLINE] (3/8/12)
- [25] INTEL *OpenCV Reference Manual*
http://www.emgu.com/wiki/index.php/Main_Page [ONLINE] (3/8/12)

Dodatok A

Obsah priloženého CD

Obsah priloženého CD nosiča je organizovaný do nasledujúcej adresárovej štruktúry:

- **/thesis**
elektronická verzia textu práce vo formáte pdf
- **/src**
zdrojové kódy projektu
- **/install**
spustiteľný inštalačný súbor
- **/audiocam**
spustiteľný program bez nutnosti inštalácie

Dodatok B

MIDI

B.1 Nástroje General MIDI 1

Piano

0	Acoustic Grand Piano
1	Bright Acoustic Piano
2	Electric Grand Piano
3	Honky-tonk Piano
4	Rhodes Piano
5	Chorused Piano
6	Harpsichord
7	Clavinet

Guitar

24	Acoustic Guitar (nylon)
25	Acoustic Guitar (steel)
26	Electric Guitar (jazz)
27	Electric Guitar (clean)
28	Electric Guitar (muted)
29	Overdriven Guitar
30	Distortion Guitar
31	Guitar Harmonics

Chromatic Percussion

8	Celesta
9	Glockenspiel
10	Music Box
11	Vibraphone
12	Marimba
13	Xylophone
14	Tubular Bells
15	Dulcimer

Bass

32	Acoustic Bass
33	Electric Bass (finger)
34	Electric Bass (pick)
35	Fretless Bass
36	Slap Bass 1
37	Slap Bass 2
38	Synth Bass 1
39	Synth Bass 2

Organ

16	Hammond Organ
17	Percussive Organ
18	Rock Organ
19	Church Organ
20	Reed Organ
21	Accordion
22	Harmonica
23	Tango Accordion

Strings

40	Violin
41	Viola
42	Cello
43	Contrabass
44	Tremolo Strings
45	Pizzicato Strings
46	Orchestral Harp
47	Timpani

Ensemble

48	String Ensemble
49	String Ensemble Slow
50	SynthStrings 1
51	SynthStrings 2
52	Choir Aahs
53	Voice Oohs
54	Synth Voice
55	Orchestra Hit

Brass

56	Trumpet
57	Trombone
58	Tuba
59	Muted Trumpet
60	French Horn
61	Brass Section
62	Synth Brass 1
63	Synth Brass 2

Reed

64	Soprano Sax
65	Alto Sax
66	Tenor Sax
67	Baritone Sax
68	Oboe
69	English Horn
70	Bassoon
71	Clarinet

Pipe

72	Piccolo
73	Flute
74	Recorder
75	Pan Flute
76	Bottle Blow
77	Shakuhachi
78	Whistle
79	Ocarina

Synth Lead

80	Square Wave
81	Sawtooth Wave
82	Synth calliope
83	Chiffer lead
84	Charang
85	Solo Voice
86	5th Sawtooth Wave
87	Bass & Lead

Synth Pad

88	Fantasia
89	Warm Pad
90	Polysynth
91	Space Voice
92	Bowed Glass
93	Metal Pad
94	Halo Pad
95	Sweep Pad

Synth Effects

96	Ice Rain
97	Soundtrack
98	Crystal
99	Atmosphere
100	Brightness
101	Goblin
102	Echo Drops
103	Star Theme

Ethnic

104	Sitar
105	Banjo
106	Shamisen
107	Koto
108	Kalimba
109	Bag Pipe
110	Fiddle
111	Shanai

Percussive

112	Tinkle Bell
113	Agogo
114	Steel Drums
115	Woodblock
116	Taiko Drum
117	Melodic Tom
118	Synth Drum
119	Reverse Cymbal

Sound Effects

120	Guitar Fret Noise
121	Breath Noise
122	Seashore
123	Bird Tweet
124	Telephone Ring
125	Helicopter
126	Applause
127	Gun Shot

B.2 Mená a čísla MIDI nôt

Fre- quency	MIDI	Inter- national	German
4186	108	C8	c ^{'''}
107			
106			
105			
104			
103			
102			
101			
100			
99			
98			
97			
96			
2093	96	C7	c ^{'''}
95			
94			
93			
92			
91			
90			
89			
88			
87			
86			
85			
1046.5	84	C6	c ^{'''}
83			
82			
81			
80			
79			
78			
77			
76			
75			
74			
73			
523.25	72	C5	c ^{''}
71			
70			
440	69	A4	a ^{''}
	68	G#4/Ab4	gis/as ^{''}
	67	G4	g ^{''}
	66	F#4/Gb4	fis/ges ^{''}
	65	F4	f ^{''}
64			
	63	E4	e ^{''}
	62	D#4/Eb4	dis/es ^{''}
	61	D4	d ^{''}
	60	C#4/Db4	cis/ges ^{''}
261.6	60	C4	c ^{''}
59			
58			
57			
56			
55			
54			
53			
52			
51			
50			
49			
130.81	48	C3	c0
47			
46			
45			
44			
43			
42			
41			
40			
39			
38			
37			
65.406	36	C2	C
35			
34			
33			
32			
31			
30			
29			
28			
27			
26			
25			
32.703	24	C1	C ^{''}
23			
22			
27.5	21	A0	A ^{''}